DCC | Digital Curation Manual

Instalment on

"Open Source for Digital Curation"

http://www.dcc.ac.uk/resource/curation-manual/chapters/open-source/

Andrew McHugh

Humanities Advanced Technology and Information Institute (HATII)

University of Glasgow, Glasgow G12 8QJ

http://www.hatii.arts.gla.ac.uk

July 2005

ver 1.6

Legal Notices

Catalogue Entry

Title	DCC Digital Curation Manual Instalment on Open Source for Digital Curation
Creator	Andrew McHugh (author)
Subject	Information Technology; Science; Technology--Philosophy; Computer Science; Digital Preservation; Digital Records; Science and the humanities.
Description	Instalment on the role of open source software within the digital curation life-cycle. Describes a range of explicit digital curation application areas for open source, some examples of existing uses of open source software, a selection of open source applications of possible interest to the digital curator, some quantifiable statistics illustrating the value of open source software and some advice and pointers for institutions planning on introducing these technologies into their own information infrastructures.
Publisher	HATII, University of Glasgow; University of Edinburgh; UKOLN, University of Bath; Council for the Central Laboratory of the Research Councils.
Contributor	Seamus Ross (editor)
Contributor	Michael Day (editor)
Date	1 August 2005 (creation)
Type	Text
Format	Adobe Portable Document Format v.1.2
Resource Identifier	ISSN 1747-1524
Language	English
Rights	© HATII, University of Glasgow

Citation Guidelines

McHugh A, (July 2005), "Open Source for Digital Curation", *DCC Digital Curation Manual*, S.Ross and M.Day (eds), Retrieved <date>, from http://www.dcc.ac.uk/resource/curation-manual/chapters/open-source/

About the DCC

The JISC-funded Digital Curation Centre (DCC) provides a focus on research into digital curation expertise and best practice for the storage, management and preservation of digital information to enable its use and re-use over time. The project represents a collaboration between the University of Edinburgh, the University of Glasgow through HATII, UKOLN at the University of Bath, and the Council of the Central Laboratory of the Research Councils (CCLRC). The DCC relies heavily on active participation and feedback from all stakeholder communities. For more information, please visit www.dcc.ac.uk. The DCC is not itself a data repository, nor does it attempt to impose policies and practices of one branch of scholarship upon another. Rather, based on insight from a vibrant research programme that addresses wider issues of data curation and long-term preservation, it will develop and offer programmes of outreach and practical services to assist those who face digital curation challenges. It also seeks to complement and contribute towards the efforts of related organisations, rather than duplicate services.

DCC - Digital Curation Manual

Editors

Peer Review Board

Preface

The Digital Curation Centre (DCC) develops and shares expertise in digital curation and makes accessible best practices in the creation, management, and preservation of digital information to enable its use and re-use over time. Among its key objectives is the development and maintenance of a world-class digital curation manual. The *DCC Digital Curation Manual* is a community-driven resource—from the selection of topics for inclusion through to peer review. The Manual is accessible from the DCC web site (http://www.dcc.ac.uk/resource/curation-manual).

Each of the sections of the *DCC Digital Curation Manual* has been designed for use in conjunction with *DCC Briefing Papers*. The briefing papers offer a high-level introduction to a specific topic; they are intended for use by senior managers. The *DCC Digital Curation Manual* instalments provide detailed and practical information aimed at digital curation practitioners. They are designed to assist data creators, curators and re-users to better understand and address the challenges they face and to fulfil the roles they play in creating, managing, and preserving digital information over time. Each instalment will place the topic on which it is focused in the context of digital curation by providing an introduction to the subject, case studies, and guidelines for best practice(s). A full list of areas that the curation manual aims to cover can be found at the DCC web site (http://www.dcc.ac.uk/resource/curation-manual/chapters). To ensure that this manual reflects new developments, discoveries, and emerging practices authors will have a chance to update their contributions annually. Initially, we anticipate that the manual will be composed of forty instalments, but as new topics emerge and older topics require more detailed coverage more might be added to the work.

To ensure that the Manual is of the highest quality, the DCC has assembled a peer review panel including a wide range of international experts in the field of digital curation to review each of its instalments and to identify newer areas that should be covered. The current membership of the Peer Review Panel is provided at the beginning of this document.

The DCC actively seeks suggestions for new topics and suggestions or feedback on completed Curation Manual instalments. Both may be sent to the editors of the *DCC Digital Curation Manual* at curation.manual@dcc.ac.uk.

Seamus Ross & Michael Day.
18 April 2005

Table of Contents

1 Executive Summary

Throughout this Curation Manual a number of individual practices, principles, techniques and technologies are suggested as being particularly appropriate throughout the digital curation life-cycle. Some are uniquely associated with issues of use and longevity, while others are more generic in their application areas, but with identifiable and significant benefits for those charged with the creation, curation and re-use of digital materials. With a range of ubiquitous advantages, open source and free software can offer tangible benefits throughout the digital curation life-cycle. By its nature the adoption of open source represents a broadly affecting cultural measure, which underpins and influences the outcome of numerous other decisions within any digital curation endeavour. Open source software is frequently available without cost, which from a management perspective facilitates the creation of digital content, and its legal status frees data creators from the onerous licensing restrictions associated with proprietary software. In terms of active use, there are rarely any kind of ongoing upgrade costs so there are fewer concerns associated with ensuring that software is maintained to the latest version. Furthermore, a range of quantifiable evidence suggests that open source applications can match and often better the performance, security and reliability of commercial alternatives. It is in the areas of long-term access and preservation that the open source approach offers some of its most relevant benefits to the digital curator. With its intrinsic transparency it is more straightforward to ensure future accessibility through migration or emulation, bereft of the legal entanglements that with proprietary commercial software may make such activities problematic. Re-use is also facilitated, with open source licenses[1] explicitly permitting the integration, alteration and redistribution of program code. Closely associated with (although by no means synonymous with) open source are open standards and open formats; both are frequently embraced by the open source community. Encoding digital information in a manner which is documented, commonly understood and not linked to an individual commercial product or intended to help pursue a corporate goal is a high priority for many open source developers and distributors.

This Curation Manual instalment[2] discusses at some lengths the relevant strengths of open source software from a digital curation perspective, as well as detailing some of its more general advantages, which must be understood in order to accept the viability of an

---

[1] For convenience and consistency, the American English spelling of the noun “license” is used throughout, since this spelling is most commonly used within discussions of this topic.

[2] This instalment adapts and builds upon materials originally published as part of "*Digicult Technology Watch Report 3*", 2005, Seamus Ross, Martin Donnelly, Milena Dobreva, Daisy Abbott, Andrew McHugh and Adam Rusbridge, http://www.digicult.info/pages/techwatch.php [Accessed: 7 April 2005, 11:30].

institutional or cultural shift to open software products. Through a series of sections it describes a range of explicit digital curation application areas for open source, some examples of existing uses of open source software, a selection of open source applications of possible interest to the digital curator, and some quantitative statistics illustrating the value of open source software.

2 Introduction and Scope

2.1 Advantages of Open Source for Digital Curators

The problems implicit in digital curation can be mitigated at every stage of a digital object's life-cycle by adopting appropriate strategies and exploiting particular technologies. The open source software movement represents and characterises a thirty-year-old software development and distribution philosophy, and offers several valuable advantages to the digital curator. In recent years, the open source ethos has come to fruition within a range of commercial and public sector environments. Core beliefs in the principle of free software availability, and of a community-based approach to software development have increasingly established free and open source software within the mainstream, where its numerous applications now reside as realistic and competitive alternatives to proprietary commercial software that has been produced within a more traditional 'behind closed doors' development model. While advantages can be identified with open source in a range of application areas, there are several intrinsic qualities that lend themselves particularly well to digital curation activities, and that make the use of open source tools an excellent starting point for data creators, curators and re-users seeking to facilitate the use and preservation of digital materials.

2.2 Proprietary Software Development and Distribution

The traditional, commercial software development model has a number of key characteristics. When a software application is created, it is written in a programming language, a human-readable syntax that broadly corresponds to the way in which a computer understands and processes information. However, for a computer to make sense of a program it has to be offered in a much 'lower level' format - ultimately the 1s and 0s of binary. In order to transform a program from human-readable form to binary, many languages require the code to undergo a process called 'compiling'. The original, or 'source' code is passed through an intermediate program and translated into computer-readable syntax, which to human eyes bears little relation to the original. Within a proprietary model, developers will typically perform the compiling process behind closed doors before distributing the binary results to customers, who can run the program and enjoy its benefits without establishing a sense of how the program works, and without any means of finding out. Users are unable to change the way the program runs, other than by using the program's inbuilt tools. Often, such utilities offer significant scope for modification – an example is the macro functionality incorporated within **Microsoft**'s *Office* suite of applications. However, *complete* control over functionality is withheld, and ultimately, changes can only be made at the

publisher or distributor's behest.

2.3 Commercial Retention of Control

Proprietary software companies are motivated by commercial concerns, and are naturally keen to strengthen their own position, often at the expense of consumer freedom. By limiting access to binary files the vendors control the functionality of their applications, and can impose limitations based on their own distribution or upgrade policies and plans. If problems occur with software, new features are sought or versions are required for alternative hardware/software platforms, they must all be negotiated with the software vendor. Similarly, the customer is quite powerless to fix bugs that are identified within the system, since to do so will generally require some familiarity or interaction with the software at source code level. Because source code access is likely to be limited to a small group of developers, changes take time to implement, and the addition of specialist functionality may be overlooked or considered commercially non-viable. From a digital curation perspective this model means that end-users are unable to identify the characteristics of the software and formats they use, and subsequently are limited in the ways in which they can inject additional functionality or preservation qualities into their digital information. Preservation strategies are likely to be hampered by legal and technical barriers related to restrictive license terms and the software's closed nature.

2.4 The Philosophy of Open Source and Free Software

In contrast, open source software is developed and released in a more transparent fashion. Instead of concentrating on the financial advantages of limiting access to source code and tightly guarding knowledge, open source software is motivated by community concerns. Source code is openly shared, contributions are welcome from competent users anywhere in the world and software is distributed free from the onerous end-user agreements that characterise a great deal of proprietary software. By empowering users with access to source code the open source methodology encourages and rewards modification, re-use, redistribution and understanding. Institutions and organisations are empowered to choose appropriate tools to achieve their intended outcomes. This helps to limit the dangers posed by relying upon specific commercial proprietary software solutions. The most obvious is the surrendering of IT infrastructure control to the commercially motivated technology vendors – often at significant costs in terms of the 'curatability' of one's digital information.

2.5 Facilitating Preservation Through Transparency

Open source technologies are no longer the marginalised preserve of bedroom hobbyists, with several open source applications among the most proven and reliable of all digital solutions. By regularly embracing the

concept of open standards, these technologies further remove the mystery from information storage and use over the longer term. As with source code availability, open standards aim to excise the opaque veneer that threatens and disrupts digital preservation, limits and curtails access to long-term stored documents, and hampers the straightforward exchange and interchange of digital content. Understanding of the structures that underlie the software and formats we use and the legal rights to recreate, modify and re-distribute these structures are great facilitators to everyone: from desktop users seeking an application specifically tailored to their needs to large-scale memory institutions that need to ensure that the software format they select to encode their digital archive will not become obsolete, unsupported and impenetrable within a few years' time.

2.6 Open Source Within the Digital Curation Life-cycle

The adoption of open source software provides several diverse benefits throughout the entire scope of the digital curation life-cycle. When determining the 'curatability' of an application or software format several important criteria must be considered. These include its longevity; the ease of its re-creation or emulation; its adherence to and use of open standards; the level of legal freedom associated with its use; its associated costs; its ubiquity; its support for metadata; and its stability. From the very conception of digital information open source presents some immediate advantages. Software acquisition costs are certainly lower than those associated with equivalent proprietary products, and although other hidden costs are involved in introducing and maintaining an open source infrastructure, several studies agree that total costs of ownership are also significantly cheaper.[3] Furthermore, transparency through source code availability and the frequent association between open source and open standards facilitates long-term comprehension and re-use, enabling creators, curators and re-users to effectively and explicitly present their digital materials alongside their underlying descriptive infrastructures. In addition, the lack of onerous licensing restrictions that accompany proprietary products and stipulate acceptable conditions for use, redistribution, transfer and reverse engineering removes many of the problems often associated with the management and redeployment of software.

---

[3] David A. Wheeler, 2005,"*Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*", http://www.dwheeler.com/oss_fs_why.html [Accessed: 7 April 2005, 11:30].

3 Background and Developments to Date

3.1 The Origins of Free and Open Source Software

The open source and free software movements share a common goal, but differ subtly in their emphases. While both oppose proprietary closed-source software development and distribution, their motivations for doing so are contrasting. Nonetheless, both movements believe in the same general ethos: that software should be made universally available in its entirety, with everyone afforded the opportunity to understand, change and re-distribute it.

The free software movement, spearheaded by the Free Software Foundation (FSF) and characterised by the writings of Richard Stallman, has at its core a predominantly political, social, and moral agenda.[4] From its origins in the late 1970s and early 1980s, the free software school grew out of frustration with the barriers imposed by the secrets and non-disclosure agreements surrounding proprietary software. Uncompromising in its philosophy, the movement argues that a number of fundamental human freedoms *depend* on the ability to access software without obstruction. The definition of free software is displayed prominently on the FSF Web pages, and can be broken down into four parts, each relating to one of four essential freedoms:[5]

1. The freedom to run a program, for any purpose;
2. The freedom to study how a program works, and adapt it to individual needs, implying access to the underlying source code;
3. The freedom to re-distribute copies;
4. The freedom to improve the program and release improvements to the public so that the whole community benefits, again implying source code access.

These simple definitions offer a comprehensive insight into the priorities and motivations of the free software movement. In the absence of a suitably unambiguous word in the English language, the classic definition is free as in free speech, not as in free beer. While there is no stipulation that free software should be made available without cost, it must be possible to re-distribute bought software at no cost if it is to qualify. Stallman refutes the traditional legal ownership arguments for proprietary software. He claims that traditional property law concepts are irrelevant since they relate to the problems caused by taking away someone else's property, not simply making a copy. According to Stallman, since programs are not consumed in the same way as other

[4] http://www.fsf.org [Accessed: 7 April 2005, 11:42]; http://www.stallman.org [Accessed: 7 April 2005, 11:30].

[5] "*The Free Software Definition*," http://www.fsf.org/licensing/essays/free-sw.html [Accessed: 7 April 2005, 11:40].

types of property, they should not be subject to the same values. The free software movement is committed to a culture whereby users can benefit from the time already spent by others solving problems, negating the need to 'reinvent the wheel' themselves.

3.2 Open Source – Free Software with Different Emphases

Some have cited the uncompromising idealism of the free software movement as one of the main reasons for its continued marginalisation within the computing community, despite its indisputably impressive track record in terms of software development. In the mid 1990s a new movement evolved in reaction to the unease often provoked by Stallman's favoured socio-political arguments. Seeking to characterise and promote the more 'sellable' aspects of free software (giving much less emphasis to the arguments favoured by the FSF), this movement was dubbed 'open source,' and is represented by the Open Source Initiative (OSI) with the programmer and writer Eric S. Raymond at its helm.[6] The OSI's foremost argument is that the unique development model underpinning free software leads to better software than that developed behind closed doors by the paid employees of commercial companies. For the purposes of this Curation Manual instalment the most notable way in which this superiority manifests itself is in terms of the increased 'curatability' of open source software. Significantly, the open source definition is not structured in terms of individual 'human freedoms', instead bearing more relation to a legal document of the kind familiar to users of commercial software products. Among its ten individual requirements are that open source software must be freely distributed; that source code must be available along with any compiled binaries; and that modifications and derived works must be permitted and re-distributable under the same license as the original software.[7]

3.3 Licensing

The most common software license under which Free and open source software is distributed is called the GNU General Public License (GPL).[8] Originally conceived to describe the legal status of the GNU operating system, this has become the generic standard free software license. It is a copyleft license, and establishes and seeks to protect the freedom of its associated software quite strictly.

Copyleft is a concept of the Free Software Foundation and serves as an alternative to traditional copyright restrictions. The coining of the term came in the light of concerns that

[6] http://www.opensource.org [Accessed: 7 April 2005, 11:42]; http://www.catb.org/~esr/ [Accessed: 7 April 2005, 11:43].

[7] Needless to say, these requirements have a great deal in common with those outlined for Free Software. However, the Free Software Foundation generally places higher ethical demands on software licenses, so while most, if not all, Free Software approved licenses will be open source, the opposite is not necessarily true.

[8] GNU is a recursive acronym for 'GNU's Not Unix'.

without some kind of protection, free and open source software could be taken by proprietary software developers, changed and then re-distributed under a proprietary non-free software license. The copyleft requirement makes it impossible to "strip off the freedom" in this fashion. Copyleft says that anyone who re-distributes the software, with or without changes, must pass along the freedom to further copy and change it. Some opponents of free software have dubbed this "a viral clause" because it means that any new software that incorporates existing copyleft code automatically inherits the same license. Thus, the code and the freedoms of the license become legally inseparable. From a digital curation perspective, copyleft offers a level of assurance that any measures taken to limit software obsolescence and to facilitate use are likely to persist within an application irrespective of any subsequent revisions or redevelopment that takes place.

One criticism that is often levelled at the General Public License is that its terms and content were conceived mainly in accordance with the United States legal system. Several commentators have suggested that the GPL is weaker or even completely inapplicable within alternative, non-US jurisdictions.[9] Consequently various regionally specific licenses have been developed. A good example is the CECILL license drafted by the French scientific research community in response to inadequacies of the GPL in the French legal context.[10] However, controversy still looms to an extent, since the Open Source Initiative is yet to approve CECILL as a conforming license. Although the intentions behind it are good, European legal reaction to it has remained somewhat wary. Several different open source and free software licenses exist. Open source licenses are those explicitly acknowledged by the OSI, and these currently number around fifty individual licenses, each with their own profile. The Free Software Foundation is responsible for identifying those licenses that qualify as free software.

3.4 Generic and Specialist Benefits of Open Source

Notwithstanding the moral and ethical arguments in favour of free and open source software, most readers will expect some insights into their more pragmatic merits before being tempted to use them, or to develop applications under their terms. There are several persuasive arguments in favour of using open source software, and of releasing under an open source license. Many of these are generic, and equally applicable to computer users in any field, activity or industry, but there are several advantages of particular relevance to the digital curation community. Clearly, a software infrastructure that facilitates digital curation can

[9] Alex Thurgood, January 2005, "*The GPL and non-U.S. law*", *Open Source Law Blog*, http://www.oslawblog.com/2005/01/gpl-and-non-us-law.html [Accessed: 7 April 2005, 11:43].

[10] http://www.cecill.info/licences/Licence_CeCILL_V1.1-US.html [Accessed: 7 April 2005, 11:43].

only be viable if it also offers functionality, value and reliability on a par with or in excess of that offered by alternative proprietary tools. The remainder of this Curation Manual instalment will therefore concentrate on both the general qualities of open source software and those aspects that make an open source software environment extremely useful from the specific perspective of digital curators.

4 How does Open Source Apply to Digital Curation?

4.1 Life-cycle as User Perspectives

In considering the value of open source for digital curation it is convenient and worthwhile to consider its merits through every stage of the data life-cycle model. Encompassing creation, active use, archiving, preservation, access and re-use, and disposal or transfer one can identify three main user roles. These are of **data creator**, **data curator** and finally **data re-user**. The following sections will consider the advantages offered by open source in the context of each of these roles.

4.2 Perspective 1 - Data Creator

4.2.1 Cost of Software

The first, and perhaps most often cited benefit of open source software relates to the issue of acquisition cost. Open source and free software need not be distributed without charge, but the definitions ensure that while a vendor could sell an open source product for a fee, its purchaser could then re-distribute it for free. Total cost of ownership is a more complex issue, incorporating a number of often more hidden costs across the entire data life-cycle, and this is explored in more depth below. Nonetheless, it can be indisputably stated that from a financial perspective, open source software empowers users to begin working with digital materials and create digital content quickly and with few onerous responsibilities.

4.2.2 Availability of Assistance

Help for data creators is also widely available within the open source community. With vast documentation projects often coexisting alongside software development, comprehensive and useful information for using and understanding open source software is increasingly available. In addition to formal help materials, an Internet-wide community of expertise contributes to an ever expanding knowledge pool consisting of discussion fora, FAQs and “how to” guides.

4.2.3 Developer Advantages

Similarly, there are numerous additional advantages in favour of creating new digital information within the open source community. From a practical perspective, a great deal of software is released under an open source license because it is the only way to legally integrate existing free software code or libraries. Because of the vast range of well-written, standards compliant and commonly understood software that is currently only available under copyleft licenses like the GPL it may be a more attractive prospect to build on the work of others than start from scratch, replicating functionality that already exists and is freely available. In addition, basing work on mainstream ‘accepted’ code automatically expands the pool of developers and users with an interest in ensuring its longevity. An open source approach also offers the opportunity to

consult and collaborate with a wide Internet-based developer community to facilitate testing and improvements. This is particularly useful for solo developers, and small groups for whom outside intervention, assistance and feedback are beneficial and otherwise unavailable. In addition, several popular web sites exist to promote and distribute open source software materials. For applications that work well, success, prominence and large-scale adoption usually follow with word spreading around the community, fuelled by exposure on sites like **Sourceforge.net**.[11] Umbrella resources like the **Open Source Technology Group** offer a platform for shared ideas and knowledge interchange, and at the same time provide mechanisms for the promotion and distribution of open source applications.[12]

4.3 Perspective 2 – Data Curator

4.3.1 Customisable Functionality

The overall depth of functionality offered by a particular program is an immediate and obvious indicator of its value from an active use perspective. Since open source involves potential users at every stage of the development process, the functionality requirements and expectations that users have can be identified and implemented effectively. Unique or marginalised functionality can be incorporated into existing applications straightforwardly, due to the availability of source code. Features that would not be worthwhile for a commercial company to implement due to the lack of overall user demand can be introduced, and new projects can be started when it seems that a particular functional requirement is unlikely to be met. The open source model empowers users to either develop their own specifically required applications or to add their own functionality to those that already exist. For example, it is possible for developers to incorporate additional digital curation functionality such as metadata support or compatibility with additional file formats into existing open source software. The culture of customisation ensures that software and digital content can be altered to suit specific requirements and expectations. The commercial software world cannot match this level of flexibility. It will often charge a fee to add a particular feature at the request of an individual client or, more likely, assure the customer that the functionality will be integrated when they pay to upgrade to the next version of the software.

Needless to say, many users of open source software are ill-equipped in terms of expertise or time to individually implement every change they require, or to affect the modifications in house. However, by facilitating development on a global basis, the open source model enables organisations to outsource freely, or to motivate others within the community who are equipped to modify or build upon code to do so. It is likely that open source users will

[11] http://sourceforge.net/ [Accessed: 7 April 2005, 11:44].

[12] http://www.ostg.com [Accessed: 7 April 2005, 11:44].

continue to seek assurances that software developers are committed to their products long-term and that ongoing maintenance will be undertaken, new features added and bugs fixed. However, although the developer-customer relationship can continue to exist in this fashion, it is not the only one that will ensure these ends are satisfied. If the developer reneges on a commitment that he or she has made then the unique status of open source software ensures that another individual or organisation can intervene and ensure the product's sustainability.

4.3.2 Peer Reviewed Software Integrity

The identification and correction of bugs from programs is another strength of the open source development model. All but the most straightforward of software programs will contain bugs: they are a regrettable, but inevitable part of software development. With traditional software development, it is common for a team of programmers to complete an application and, through a period of evaluation, to identify and fix errors. However, software companies face great pressure to get their products on the shelves to begin generating income, and therefore bug-fixing schedules are often necessarily limited. Users will often find flaws, but without access to source code it is impossible for them to personally remedy these; the only recourse is to notify the relevant software publisher. If a bug is sufficiently serious then the company will probably issue a software patch to repair the flaw. However, since they must rely on error reports from users with no access to source code it can be difficult to trace bugs; this represents a failure to exploit the application users' programming and debugging abilities and dramatically lengthens the process. In addition, there is usually nothing but goodwill to guarantee that companies make any corrections available free of charge and they might simply refuse to address the flaw at all, leaving users with no option but to learn to accept deficiencies in their applications.

Relying on the philosophy of releasing software early and often, open source projects are likely to receive users' bug reports before a program is even close to the level of maturity that a commercial company would deem acceptable for release. With access to source code, collaborators can fix problems themselves, or offer detailed accounts in programming terminology of where and in what circumstances bugs manifest themselves. "Treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging," writes Eric Raymond, his mantra: "Given enough eyeballs, all bugs are shallow."[13] In addition, open source projects are becoming increasingly well documented, with the rapid growth and mainstream proliferation of open source leading to the generation and prioritisation of good quality documentation. As well as facilitating debugging the peer review system can also be used to ensure that digital information or content is sufficiently

[13] Eric S. Raymond, "*The Cathedral and the Bazaar*", http://ot.op.org/cathedral-bazaar.html [Accessed: 7 April 2005, 16:28]

functionally rich throughout its development and maintenance. For instance, from a digital curation perspective, collaboration can take place to ensure that the information infrastructures are optimised for longevity, continued accessibility and re-use. While poorly written code is by no means the exclusive preserve of the proprietary software development world, the chances of a bad open source program continuing to be developed badly are mitigated somewhat by the community's watchful eyes.

4.3.3 Users Assume a Strong Legal Position

An argument often raised by those opposed to open source is that if something goes wrong with the software there is no one to direct the blame towards. From the digital curator's point of view this might provoke concerns: if one relies upon a particular software package or data format to ensure the curatability of digital resources then there are certainly grounds for dissatisfaction and an expectation of recompense if this is not achieved. While this is true, the majority of proprietary licenses will include terms absolving responsibility for problems caused by flaws or shortcomings in software. For instance, no one has successfully sued **Microsoft** for downtime or information lost as a direct consequence of security loopholes in their *Windows* operating system.[14] Due to the financial models underpinning a variety of open source software it is likely that vendors will offer support contracts and software guarantees as a commercial service quite distinct from the distribution of software itself, which may incorporate rights to compensation in the event of a failure to meet their commitments. In addition, as discussed above, the transparency of open source enables individual developers or administrators to independently implement solutions to overcome the shortcomings in the programs that they use.

4.3.4 Increased Security of Digital Resources

The Internet remains a dangerous place, with the potential for virus infection, denial of service attacks and interception of personal details presenting serious concerns. Security is therefore something that must be taken into consideration during any digital curation work flow. For instance, where materials are stored in remote repositories, security must be assured in order to be confident that retrieved information has not been compromised, or altered from its initially deposited form. It is often argued that by making source code available it will be easier for malicious individuals to identify and exploit security vulnerabilities in open source software. This is dismissed by open source advocates as a false argument, and typical of the "Fear, Uncertainty and Doubt" strategies frequently

[14] In fact, Microsoft's Windows XP license includes the clause "In no event shall Microsoft…be liable for any…damages whatsoever…even in the event of fault…(including negligence)." Todd Bishop, September 2003, "*Should Microsoft Be Liable For Bugs?*" *Seattlepi.com*, http://seattlepi.nwsource.com/business/139286_msftliability12.html [Accessed: 7 April 2005, 11:45].

employed by the proprietary software world. The sense of 'security through obscurity' promoted by proprietary software companies is considered to be rather dangerous: not only does it create a false sense of safety, but it also limits opportunities for identifying existing security loopholes. Malicious 'crackers' are motivated and determined, and will uncover any vulnerabilities that exist whether code is freely available or not. Opening the source ensures that those who are interested in finding problems to fix and secure (rather than to exploit) can do so as straightforwardly as possible. Also, while it should certainly not lead to complacency, open source users are less likely to find themselves the target of malicious attacks than those using proprietary tools, since a great deal of destructive code is motivated by distrust and resentment directed towards large corporations. This situation could change if open source continues to establish itself within the mainstream, since the intention of malicious crackers may be to simply attack the biggest targets, irrespective of political factors.

4.4 Perspective 3 – Data Re-users

4.4.1 Longevity of Digital Information

The expected lifetime of open source software compares favourably with proprietary alternatives, although arguments can be presented to suggest that either is assured of greater longevity. Ubiquity is an appealing characteristic, and it is frequently maintained that mainstream commercial applications with large distributions are more likely to be accessible in the future due to the sheer number of people who have a vested interest in ensuring that this is the case. Few would question the fact that increasing the number of stakeholders is likely to increase the demand for an application or file format's sustained accessibility (the so-called 'follow the crowd' approach). Particular concerns could be levelled at open source projects that are marginalised within the overall digital community, and formats that although open are less well supported and less frequently use than commercial alternatives, such as the *OpenOffice 1.1* document or *Ogg Vorbis* digital audio formats. However, one must make a clear distinction between the size of the user community that is interested in ensuring an application or data-set's longevity and the straightforwardness with which this can be achieved. Digital curators face both sociological and technical challenges. It is suggested that the latter are better addressed by the use of open source. Transparency is at the very foundation of open source and free software, promoting understanding and facilitating its curation. In addition, such software is far more likely to embrace open formats and standards in favour of proprietary alternatives. Therefore, although there may be more voices demanding the curation of commercially distributed proprietary software, it is likely that a comparatively modest number of open source users can achieve the same goal with less effort and with significantly less expense incurred. People power can make it easier to overcome the

barriers to digital curation, but although currently less well used, open source software itself overcomes a number of the most problematic obstructions that the digital curator is likely to face.[15] Furthermore, while it lacks comparable user numbers in many disciplines and application areas (particularly the desktop domain), there are several areas in which open source software, such as the *Apache Web Server*, the *Bind* DNS server and prominent institutional repository implementations like *DSPACE*, *Fedora* and *GNU EPrints* hold dominant positions, even over commercial alternatives.

4.4.2 The Relationship Between Open Source and Open Standards

The philosophies underpinning open source software have close associations with the concepts of open standards that are vital for successful exchange, re-use and preservation of documents and data. Open standards are those that, by virtue of their transparency and accepted nature, offer a degree of protection against obsolescence and inaccessibility. Technologist Bruce Perens suggests a definition of the principles and practices surrounding open standards, and offers detailed insights into what significant details elevate a common specification to the status of open standard.[16]

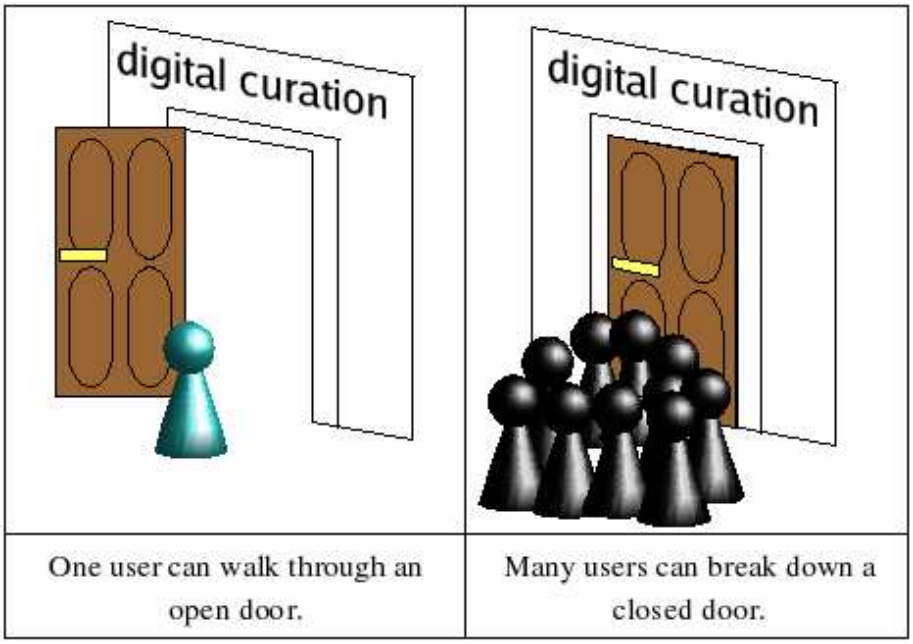

Figure 1
© Digital Curation Centre

[15] See Figure 1

[16] Bruce Perens, "*Open Standards: Principles and Practice*" http://perens.com/OpenStandards/Definition.html

According to Perens' criteria an open standard offers the freedom to view and implement it, prevents customers from being 'locked in' to a particular vendor or group, and ensures that there is no associated royalty or fee and no favouring of one implementer over another. Although it should be possible to extend open standards or offer them in subset form controls must exist to prevent dominant vendors from implementing the standard with extensions that are incompatible with other systems. Close parallels can be drawn between these principles and those expressed within the open source definition, particularly in terms of the concepts of freeness and unencumbered total access that each promotes. Examples of open standards for file formats include the OASIS Open Document Format, the World Wide Web Consortium's (X)HTML and **Adobe's** PDF format. These represent reasonably safe starting points for storing content for future retrieval since all are understood, documented and published. There are no associated licensing costs and no charge can be levied for their use and distribution. Commercial software companies tend to assume that if they can propagate their own particular file formats widely enough, people will soon become reliant upon them. The most obvious example is **Microsoft's** *Office* suite, which uses the core file formats *.doc*, *.xls*, *.mdb* and *.ppt* to encode word-processed documents, spreadsheets, database files and slide show presentations. None of these are open standards, and therefore it is impossible to gain a thorough understanding of how they work. Consequently there is no way to confidently read and write to these formats with programs other than **Microsoft's** own.[17] Recent reports from Microsoft suggest that future versions of their document formats will be defined in XML, which should theoretically introduce a greater degree of transparency in their structure. However, within a community suspicious of **Microsoft** (following for instance their extremely limited 'Shared Source' scheme[18]) few expect these plans to lessen the opacity intrinsic to **Microsoft**'s products to any significant degree. These expectations are galvanised with **Microsoft**'s failure to offer confirmation that future versions of their software will support OASIS's Open Document Format. David Rosenthal argues that the

[Accessed: 7 April 2005, 11:46].

[17] Numerous projects, such as *OpenOffice.org* have tried to remedy this, with some success. See also summary of CAMiLEON working papers in: *The DigiCULT Report Full Report* "Technological Landscapes for tomorrow's cultural economy: Unlocking the value of cultural heritage", (January 2002), p. 212. Available online at http://www.digicult.info/pages/report.php, [Accessed: 7 April 2005, 11:46]

[18] 'Shared Source' is a Microsoft initiative under which enterprise users, academics and others can get controlled access to select parts of Microsoft's source code. Heavily criticised for its toe-in-the-water conservatism, The UK Register web site described it as nothing more than a "worthless PR exercise", Andrew Orlowski, 2004, "*Why Microsoft 'Shared Source' Can Never Be Trusted*", http://www.theregister.co.uk/2004/03/17/why_microsoft_shared_source_can/, [Accessed: 6 July 2005, 15:30]

incompatibly of data formats within the *Office* suite is a deliberate and quite integral part of **Microsoft**'s business model. By distributing its software at low cost with new computers pressure is placed on users to pay for expensive 'essential' upgrades that are subsequently introduced. The case for upgrading can be persuasive – ongoing support may depend on it and poor backwards compatibility in many products may render collaboration impossible if one's peers are running a newer version. The implications that this has for accessibility need little further explanation. By its nature open source software can be understood and if necessary replicated at a later date, and open formats boast similar long-term lucidity. Developers and users can continue to access open digital resources in the future more straightforwardly than proprietary assets with missing digital jigsaw pieces.

4.4.3 Portability of Information

In terms of ease of emulation and potential portability open source carries a significant advantage. Without having to painstakingly reverse engineer existing applications, software environments and data formats it is theoretically straightforward to take information in a particular form or structure and recreate or repackage it as required.[19] Binary-only, non-documented and non-standard software is shrouded in mystery, with even sophisticated decompiler software unable to offer reliable and definitive insights into fundamental underlying qualities. The efforts of *OpenOffice.org* to create a standards compliant productivity suite supporting a range of both proprietary and open formats offers a somewhat trite, but insightful example of the kinds of barriers faced when dealing with commercially encoded digital assets. The contemporary problems associated with these are likely to be amplified many times in the future. Without an intimate understanding of the **Microsoft** *Word* format for instance, it is impossible to adequately and confidently render all the information contained within a *.doc* file in any non-**Microsoft** endorsed environment.[20] As long as one has to rely upon an individual private corporate organisation to access and use one's digital content, it can never be effectively curated, and its longevity can never be assured. Technology journalist David Berlind pulls no punches: "Putting the vendor in control of your IT costs is not a good position to be in. Unfortunately, that's where a lot of us are."[21]

---

[19] S. Ross and A. Gow, 1999, "*Digital archaeology? Rescuing Neglected or Damaged Data Resources*", (London & Bristol: British Library and Joint Information Systems Committee), ISBN 1900508516, http://www.ukoln.ac.uk/services/elib/papers/supportin g/pdf/p2.pdf [Accessed: 7 April 2005, 11:47].

[20] Maria Guercio and Cinzia Cappiello, 2004, "*File Formats Typology and Registries for digital preservation*", (DELOS, WP6 D6.3.1), http://www.dpc.delos.info [Accessed 7 April 2005, 11:48].

[21] David Berlind, July 2002, "*Who Gave Microsoft Control of Your IT Costs? You did*", http://techupdate.zdnet.com/techupdate/stories/main/0, 14179,2875958,00.html [Accessed 7 April 2005, 11:48].

These costs are likely to be more than simply financial. The combination of proprietary software and formats puts the software distributor in an unhealthily powerful position, and exposes the customer to the cost of 'essential' upgrades and even greater problems should the technology be discontinued or the developer become insolvent. Every time that a user records data in a closed format it tightens the grip held by its proprietary developer. The chain becomes increasingly more difficult and more expensive to break away from. It should be clear that when preservation issues and future content access are considered, the problem is only exacerbated. The OAIS reference model speaks of the importance of the availability of Representation Information to ensure that the informational value of our preserved bit-streams remains available even into the future. Among the most frequently cited items of Representation Information include software and format specifications, which with an open approach to software management, acquisition and distribution are likely to be much more readily available than within a proprietary commercial model.

4.4.4 Preservation Through Transparency

The digital curation community is no stranger to costly projects that have failed because of technological choices that were overly proprietary or marginalised. The **BBC's** Domesday project is perhaps the most frequently cited example.[22] In 1986, to celebrate the 900th anniversary of the Domesday Book, a project was undertaken to incorporate a diverse range of materials contributed by UK schoolchildren within a multimedia resource. Unfortunately, the project's technological choices led to certain subsequent problems. For storage media, the project team chose to use **Philips**' proprietary LaserVision LVROM disc, which could only be played on the associated LVROM player. The multimedia application itself was written in a language called *BCPL*, a precursor to *C* which ran on the **BBC** Model B platform, which had to be modified to interface with the proprietary discs, increasing costs and limiting the chances of viability and uptake. Regrettably the system soon became obsolete and, less than twenty years later, very few players or discs remain. Only the sustained efforts of the CAMiLEON project to rescue the application and implement an emulation strategy have ensured that future generations can access this valuable resource.[23]

Domesday need not have come up against such problems if a more future-conscious series of decisions had been made at its conception. One of the biggest single issues for the project (and for digital curation more generally) was its inability to ensure that at an unknown time in the future users would still be able to access the

[22] http://www.atsf.co.uk/dottext/domesday.html [Accessed 7 April 2005, 11:48].

[23] http://www.si.umich.edu/CAMILEON/domesday/domesday.html [Accessed 7 April 2005, 11:49]. See Daisy Abbott, "*Overcoming the Dangers of Technological Obsolescence: Rescuing the BBC Domesday Project*", *DigiCULT.Info 4*, Page 4 , http://www.digicult.info/pages/newsletter.php [Accessed 7 April 2005, 11:49].

stored digital materials.[24] An analogy with the original Domesday Book is offered on the CAMiLEON project web site. If the Latin language in which the original Domesday Book was written became somehow incomprehensible, accessing the information it holds would be impossible.[25] A Latin dictionary could be used to overcome this problem and the OAIS reference model would call this Representation Information. What must be remembered is that the remit of digital curator is not limited to simply maintaining the physical materials themselves, something which is conceptually quite straightforward; it is also necessary to ensure that a method of understanding and exploiting their full usefulness continues to exist in perpetuity. Instead of dwelling on the curation or preservation of data one must strive for long-term access to *information*. By using open, standardised formats, one can more feasibly limit the problems caused by the passage of time. If the Domesday project had used an open standardised structure to describe and encode its multimedia components, together with open formats for incorporated sound and video, it is likely that a clearer understanding of the data structures could be established in the future, with less guesswork or painstaking reverse engineering procedures. Furthermore, if the source code of the application was made freely available it too could continue to be understood and broken down into more easily migrated algorithmic chunks.

New digital hardware will inevitably be introduced, and it is likely that the machines we use ten years from now will operate quite differently from those in we are familiar with today. But the hardware level is just one of several potential areas where problems can occur for the digital curator. Assuming that data are encoded in an open file format, and that the programs that read, access and write to these formats are open source, any hardware-related preservation problems can be more straightforwardly overcome. The knowledge conferred by the use of open source applications and open standards empowers future users, enabling resources to be more straightforwardly manipulated within a future hardware configuration. The INFORM methodology proposed by Andreas Stanescu[26] suggests several classes of risk, including those originating from the digital object's format, its associated software, and organisations and communities related to the preservation plans for the object. Open source software and open, standardised formats are likely to fare very well

[24] For an example of an open source tool which 'normalises' file formats for preservation see Adam Rusbridge, April 2004, "*XENA: Electronic Normalising Tool*", *DigiCULT.Info*, Issue 7, page 32, http://www.digicult.info/pages/newsletter.php [Accessed 7 April 2005, 11:49].

[25] http://www.si.umich.edu/CAMILEON/domesday/faq.html, Accessed 7 April 2005, 11:50

[26] Andreas Stanescu, 2005, *"Assessing the durability of formats in a digital preservation environment: The INFORM methodology"*, (OCLC Systems and Services, International Digital Library Perspectives, Vol 21, Number 1, 2005, pp. 61-81)

in each of these categories. In a closed source environment, future preservationists will face the onerous tasks of reverse engineering software to run on new platforms, developing emulators without a clear understanding of systems that need to be replicated and continuing to maintain otherwise obsolete hardware upon which resources are known to operate.

4.4.5 Legal Issues for Long-term Access

The practical problems inherent in dealing with proprietary software represent only part of the problem. In addition there are likely to be legal obstacles to the emulation or porting of commercially distributed, proprietary software. Terms of use, usually strictly defined in software license agreements generally operate on a can-do basis, with an implicit assumption that if a particular type of use is not mentioned that it is forbidden. It is therefore common for many commercial software licenses to prohibit the emulation, porting, migration and reverse engineering of application information or datasets. Similarly, restrictive terms of use may infringe upon one's ability to collect and maintain appropriate Representation Information to prolong the useful life of a digital object. For software to qualify as open source or free however, there are assumptions to the contrary, in favour of freedom of use, re-use and redistribution. License agreements associated with OSS make it easier to take preservation measures without fear of violating the intellectual property claims of the original developers. While proprietary vendors may have little interest in continuing to support their software indefinitely and seldom offer the means to ensure its longevity, they often have a tendency to legally challenge anyone else who attempts to do so. This places a further burden on the user to administer licensing and terms of use documentation to ensure that the correct infrastructure is in place and that no infringements can take place. Open source has no such problems – in contrast, only steps taken to limit free access to open source software are likely to fall foul of licensing agreements.

4.4.6 Later Stages

Later stages of the digital curation lifecycle, such as disposal or transfer of stewardship are further facilitated and simplified within an open source infrastructure. With none of the legal barriers to redistribution that are often explicitly forbidden under proprietary licenses open source materials are are comparatively straightforward to disseminate and transfer.

5 Open Source and Free Software In Action

5.1 Areas of Use

From its origins in laboratories, technology centres and student dormitories, open source has exploded in popularity over the last few years, and now performs a key role in the IT policy and infrastructure of many organisations, institutions and companies. Various reasons are cited for the adoption of open source, and these tend to vary across the sectors in which it enjoys exposure. Many users are attracted to the traditional stability and reliability of the software, others to its straightforward integration with heterogeneous system environments, and others, notably the digital curation community, to the empowering transparency and freedom that are both intrinsic to open source. Many more may be convinced by the financial savings that might be achieved from using these technologies.

5.1.1 Government and Public Sector

The increasing success of open source in the public and government sector has been one of the more significant developments of recent times in terms of technology take-up. Public sparring between proprietary software companies and the open source movement for lucrative governmental IT contracts emphasises the significance of this market, particularly for the subsequent dissemination of technologies throughout the public and social hierarchy, within the new era of e-government and digital public administration. Numerous reasons can be identified to explain the enthusiasm with which open source has been embraced by many public bodies. Perhaps the most obvious are related to the financial savings it affords, which in governmental terms may be a vote-winner. However, this is only part of the story. There can be little doubt that government bodies and agencies are to some extent wary about the potential consequences of trusting their entire IT infrastructure to one or two private (usually foreign) companies who are likely to guard their software secrets closely. Open source software is able to nullify this problem. In addition, governments are invariably charged with the responsibility of ensuring that provisions are in place to preserve a wide range of information for future generations. Open source facilitates this in a way that proprietary infrastructures cannot. Governments are formally charged by their electorate with the responsibility to maintain public records long-term, and legislation such as the Freedom of Information Act in the United Kingdom offers persuasive arguments for a move towards a more open data environment.

There are numerous examples of large-scale public sector migrations to open source within Europe and further afield. The French Government has decided that central administration should terminate most of its agreements with proprietary vendors for the supply and use of software, meaning that

national and local authorities are to use open source software as far as possible. The Agency for Information and Communication Technologies in Administration (ATICA) was set up in August 2001 to support this decision, and to coordinate the various governmental agencies and bodies towards the intended outcome. Similarly, the German central administration in June 2002 entered into a framework agreement with **IBM** and **SUSE** on the supply of open source products based on *Linux*, making it possible for German public administration to acquire *Linux*-based systems at a reduced price from **IBM**.[27] The agreement incorporates the supply of servers and workstations, as well as ongoing support from **IBM**. While this promotion of open source is not a law, it represents a tempting incentive to open source decision-makers within the German public sectors. The United Kingdom has shown a clear commitment too, and the British Office of E-envoy issued an open source policy at the end of October 2004 which states that British Government and authorities will in future consider open source, declaring in particular a concern about being 'locked-in' to the products of single private commercial companies.[28]

The International Institute of Infonomics report entitled 'Free/Libre and Open Source Software: Survey and Study' (2002) recommended and reported a widespread deployment of open source tools throughout European government.[29] This document contains several accounts of the public sector embracing open source systems. The French Ministry of Culture migrated 400 servers from *Unix* and *Windows NT* to *Linux* and intends to have comprehensive *Linux* server solutions by 2005. The Ministry of Justice and national crime register use a combination of open source tools such as the *Apache Web Server*, *Perl*, *Samba*, and *fetchmail*, with an imminent migration envisaged from proprietary *Unix* to *Linux*, *PHP*, and *MySQL* and finally the Ministry of Defence have *FreeBSD*, an open source operating system comparable to *GNU/Linux*, installed within their infrastructure.

In what is regarded as one of the most significant developments in the lifetime of open source software, the City of Munich in Germany officially confirmed in June 2004 that it would be transferring 14,000 municipal desktop computers from **Microsoft** *Windows* to open source, combining *Linux* server software, desktop software, and virtual machine technology from *VMware* to provide

---

[27] http://www.ibm.com [Accessed 7 April 2005, 11:50]; http://www.suse.com [Accessed 7 April 2005, 11:51].

[28] See Danish Board of Technology, October 2002, "*Open Source Software in e-government*", http://www.tekno.dk/pdf/projekter/p03_opensource_paper_english.pdf [Accessed: 7 April 2005, 11:51] and UK Government, October 2004, "*Open Source*", http://www.govtalk.gov.uk/policydocs/policydocs_document.asp?docnum=905 [Accessed: 8 July 2005, 16:27]

[29] Except where otherwise stated, accounts are from International Institute of Infonomics, 2002, "*Free/Libre and Open Source Software: Survey and Study",* http://www.infonomics.nl/FLOSS/report/ [Accessed: 7 April 2005, 11:51].

interoperability among heterogeneous systems.[30] Bloomberg News described this as **Microsoft's** "biggest PC loss yet," and the decision has led analysts to predict that *Linux*-powered PCs will grow 25%-30% in 2004, and that *Linux* will account for 6% of desktop operating system shipments by 2007.[31] Bergen, Norway's second city, has followed in the footsteps of Munich, Germany in choosing *Linux* to underpin its technology infrastructure, moving away from proprietary UNIX and **Microsoft** *Windows* platforms and applications.

5.1.2 Humanities Institutions

Like public sector institutions, the cultural heritage sector has a vested interest in both the financial cost and the openness and accessibility of the software it uses. Many institutions have discovered that open source solutions offer advantages to facilitate their requirements. A prominent example is the National Library of Australia, which now deploys a range of open source applications across its server space, reflecting a willingness to invest in the skills of the library and a commitment to standardisation in general.[32] The Library's Director of IT Business Systems, Mark Corbould, described how the institution has hesitated to replace its 700 proprietary workstations with open source however, since "*Windows* is so entrenched in the desktop space that it would take a nuclear war to remove it."[33] This seems to be an argument in favour of change sooner rather than later, and could be read as a firm assertion of the difficulties posed to effective digital curation by the current proprietary configuration.

5.1.3 Science

With bleeding edge innovation evident throughout every scientific discipline, it is unsurprising that software developed within the open source model has been embraced wholeheartedly by the science community. Significant institutions and organisations such as NASA have displayed a commitment to distributing their endeavours under open source licenses, with the NASA Open Source Agreement[34] conceived as a license determining legal usage for a range of applications. Relevant projects include artificial intelligence software systems (*Livingstone2*), dynamic 3-D world environments (*World Wind*), a simulation toolkit for planetary exploration vehicles (the *Mission Simulation Toolkit*) and a evolutionary simulation (*JavaGenes*). NASA cites four main motivators for their adoption of open source technologies and release habits. Increasing

[30] http://www.vmware.com/, [Accessed: 7 April 2005, 11:53]

[31] June 2004, "*Munich Linux Decision Fina*l", DesktopLinux.com, http://www.desktoplinux.com/news/NS7137390752.html [Accessed: 7 April 2005, 11:55].

[32] http://www.nla.gov.au/ [Accessed 7 April 2005, 11:56].

[33] Nadia Cameron, September 2003, "*Open Source Bookmarks Australian Heritage*", http://www.computerworld.com.au/index.php?id=522130461&fp=16&fpid=0 [Accessed 7 April 2005, 11:56].

[34] http://www.opensource.org/licenses/nasa1.3.php [Accessed 7 April 2005, 11:56].

software quality via community peer review, accelerating development via community contributions, maximising the awareness and impact of NASA research and increasing dissemination of NASA software in support of the education mission are together identified as sufficiently worthwhile ends to justify utilising the open source development model.

A range of scientific applications available under open source licenses are made available by projects like *OpenScience*, which gathers a diverse selection of applications intended to facilitate the work of various communities. Examples include tools for conducting studies of forensics, acoustics, astronomy, life sciences, nanotechnology and chemistry. In addition, several initiatives exist to promote the open source ethos more generally throughout the sciences. BIOS (Biological Innovation for Open Society) aims to "extend the metaphor and concepts of open source and distributive innovation to biotechnology and other forms of innovation in biology" and to facilitate "the cooperative invention, improvement and sharing of biological technologies".[35] Taking a sceptical view about the wide proliferation of restrictive patents within biological sciences, the initiative has identified a requirement for more transparency to encourage knowledge dissemination and the progression of long-term communities of understanding. Other work like Science Commons,[36] an off-shoot of Creative Commons[37] carries similar emphases, with its intention to promote innovation through knowledge sharing. In addition to developing open source software and promoting its ideals the science community has exhibited a consistent enthusiasm for using existing open source tools. A good example is the NASA Acquisition Internet Service, which in 2000 was moved without a hitch to the open source MySQL database, which has continued to provide a robust foundation to this service ever since.[38]

5.1.4 HE/FE Institutions

Notwithstanding the benefits of open source from a digital curation perspective, Richard Stallman expresses a passionate belief that all educational institutions should use free software for several additional reasons. He cites the financial savings, moral influence, and additional learning opportunities that source code availability affords as major motivators.[39] Many schools and universities have responded to these and other justifications by implementing open source solutions within their IT environments. Projects such as OSS Watch, funded by the UK's Joint Information Systems

---

35 http://www.bios.net/daisy/bios/15 [Accessed: 7 April 2005, 11:57].

36 http://science.creativecommons.org [Accessed: 7 April 2005, 11:57].

37 http://creativecommons.org/ [Accessed: 7 April 2005, 11:57].

38 Paula Shaka Trimble, December 2000, "*Open Minds on Open Source*", *FCW.com*, http://www.fcw.com/fcw/articles/2000/1204/pol-nasa-12-04-00.asp [Accessed: 7 April 2005, 11:57].

39 http://www.gnu.org/philosophy/schools.html [Accessed: 7 April 2005, 11:57].

Committee (JISC), offer advice, support, and expertise to higher and further education institutions interested in deploying open source solutions.[40] An Insight Special Report entitled 'Why Europe Needs Free and Open Source Software and Content in Schools' indicates areas in education in which open source can be deployed. The report concludes that OSS provides "a beneficial way to transfer knowledge and best practice."[41]

A recent study among IT specialists in thirty-seven tertiary education institutions in the UK and Antipodes showed that free and open source software is already in place in 94% of surveyed institutions.[42] A number of commonly used Virtual Learning Environment packages (VLEs) are open source, including the popular *moodle*,[43] designed to facilitate the creation of online courses.[44] Teaching and learning materials, which represent some of the most valuable resources generated in Higher and Further Education institutions, can be more effectively managed and maintained within such open source infrastructures.

Developers face a range of difficulties in Higher and Further Education institutions where intellectual property fruits of employees' research activities are often retained by the institution itself. In such cases decisions about redistributing the IPR rest with the owner. In such cases it is vital that employees and researchers familiarise themselves with the terms of their employment to ensure that their participation in open source development is legitimate and acceptable. The copyleft requirement within many open source licenses compels those who adapt, build upon or integrate the licensed code to release the fruits under the same license. Therefore, it is vital that employees are aware of the implications within their own institution of utilising copylefted open source products.

5.1.5 Commercial Organisations

Identifying the quality of software and the financial savings available, the enthusiasm with which some corporations have embraced open source is indisputable.[45] Prevalent now in a range of often mission-critical applications, open source performs a range of roles, from the generic to highly specialist within both small to

[40] http://www.jisc.ac.uk/ [Accessed: 7 April 2005, 11:57]; http://www.oss-watch.ac.uk/ [Accessed: 7 April 2005, 11:57].

[41] http://www.eun.org/insight-pdf/special_reports/Why_Europe_needs_foss_Insight_2004.pdf [Accessed: 7 April 2005, 11:57].

[42] David G. Glance, Jeremy Kerr and Alex Reid, January 2004, "*Factors Affecting the Use of Open Source Software in Tertiary Education Institutions*", http://www.firstmonday.org/issues/issue9_2/glance/ [Accessed: 7 April 2005, 11:57].

[43] http://moodle.org, [Accessed: 7 April 2005, 11:57]

[44] Open source can also be used for content management in educational institutions. See Paul Conway, December 2003, "*Zope at Duke University: Open Source Content Management in a Higher Education Context*", *DigiCULT.Info*, Issue 6, p. 10, http://www.digicult.info/pages/newsletter.php [Accessed: 7 April 2005, 11:57].

[45] IBM is a good example. See http://www-136.ibm.com/developerworks/opensource/, [Accessed: 7 April 2005], 11:57 for more details.

medium enterprises and multi-national mega-corporations. Notable example of companies with open source deployments or interests include **IBM**, **Novell**, **Hewlett Packard** and **Yahoo**! These deployments range from web servers and database infrastructures, to large-scale distributed computing projects. Facilitating commercial interactions and the management of corporate information long-term, open source and open standards are likely to continue to perform a vital role within the commercial sector.

5.2 Open Source Applications

Since the early development of the GNU/Linux operating system, the open source software library has grown at an impressive rate. From an initial emphasis on server and software infrastructure code, an increasing number of open source projects now commit their resources and efforts to the development of desktop applications in a range of areas, including general office productivity, multimedia development, sound and video editing and manipulation, scientific analysis and desktop publishing. As wide-ranging as open source software is in terms of functionality, it also varies greatly in terms of maturity, stability and performance. Given the vast array of projects currently at different stages of development, identifying the most valuable, useful or technologically worthwhile applications can be difficult. Similarly, with such a broad range of software, locating a particular application can be intimidating, despite the range of excellent web and repository search tools currently available.

These problems can be addressed using a number of open source resources. Two of the most prominent web-based examples – **SourceForge** and **Freshmeat** – serve distinct but similar functions.[46] **SourceForge** provides free hosting and Web space for thousands of individual open source projects, offering centralised search tools, distribution across several worldwide mirrors and a large community of users offering advice, feedback and impressions of software projects. **Freshmeat** essentially comprises a massive index of "preferably" open source applications and tools for a range of platforms, together with links to each project's own pages where the software itself can normally be downloaded. Popularity details, ratings and vitality statistics are maintained and presented, offering novice users clear insights into the success and level of use of individual applications. With an application's ubiquity offering one insight into its curatability, it is important to be able to identify just which are the most used applications and software formats.

Simply browsing these impressive resources offers insights into the range of open source tools that exist, as well as the diversity of the applications areas that are covered. In the realms of infrastructure, server and development

[46] http://sourceforge.net [Accessed: 7 April 2005, 11:57]; http://freshmeat.net [Accessed: 7 April 2005, 11:57]

software, several programs are commonly held to be as good as or better than their proprietary alternatives. Open source software like the *Linux* Operating System, *Apache Web Server* and *Bind* DNS Server, combined with a range of open standards, are all integral parts of the Internet, and one cannot overstate the role that the concept of openness played during the Web's conception. Recent times have seen an increasing number of more mainstream desktop applications move towards this level of excellence. In addition to continuing to refine the usability and functionality of general desktop applications, it is also a priority for the open source world to expand to meet the functional requirements of more specialist users. To this end, a number of excellent applications specifically aimed at the field of digital curation are now available under open source licenses.

A concern that is often vocalised is that the open source development model, based as it is upon the concept of collaboration, is likely to result in monolithic software infrastructures, with single choices that represent a compromised community consensus. In many cases there is some truth in this. 'Forking' is a term that describes the process of branching the development of source code over two or more separate, perhaps incompatible paths: this is strongly discouraged within the open source community. Many open source advocates will argue that when several individual alternative projects condense into a single unified effort it is a good thing, since although competition within the software industry is good for business, it doesn't necessarily lead to the development of the best software. Since the open source model pools expertise and doesn't set programmers the task of usurping one another it can achieve a great deal quite quickly. However, this argument is not quite sufficient to quash these concerns. Diversity is welcome within software to overcome systemic failures when they arise. As in many disciplines, mistakes are made, and spreading the intellectual effort more widely is likely to ensure the non-fatality of any problems that are encountered. Nonetheless, although a great deal of open source work finds itself expressed in just one or two applications within each domain area there still exists some welcome diversity. For instance, as the examples below illustrate, there are a number of individual and distinct institutional repository implementations currently being developed and released under open source licenses.

5.2.1 The *GNU/Linux* Operating System

The operating system is the central software program within any computer system, communicating at a low level with the microprocessor and other hardware, and organising the execution and run-time of each installed program. Among the most common and familiar examples of proprietary operating systems within the personal computer market are **Microsoft's** *Windows* and **Apple's** *OS X*. The free software movement would have had no foundations if it had had to rely upon a central proprietary program, and this realisation led to the conception of the *GNU* (GNU's not Unix)

project in 1984, to develop a suite of applications that would represent a free operating system. Identifying early on that multi-platform support was a desirable characteristic, Stallman chose to base his system on the widely published core concepts that are shared by *Unix* computer systems, the only platform at the time to offer a degree of portability. Following success with a number of applications, eventually *GNU* lacked only a kernel to make it into a complete, fully functional operating system. The kernel represents the heart of the operating system and manages system memory, the file system and disk operations. In a timely coincidence, a young Finnish programmer, Linus Torvalds, was concurrently building his own *Unix* compatible kernel, *Linux*, and this was swiftly integrated into the existing *GNU* code, resulting in what is now known as *GNU/Linux*.[47]

Since its initial release in the early 1990s, *GNU/Linux* has undergone almost constant refinement, and now represents a mature, stable and usable platform, incorporating most of the features of expensive proprietary *Unices*, and representing a viable solution for both server and desktop deployment.[48] A number of companies distribute the system with straightforward installation packages and a variety of fully integrated applications. Some of the most popular 'distributions' include ***Red Hat***, ***SUSE***, ***Mandrake*** and ***Debian***.[49] Each of these can be downloaded for free from its associated web site or purchased on CD or DVD for a small sum, fully packaged and documented. Most distributions also offer corporate packages, with full support structures more akin to commercial proprietary systems.

Along with hardware, the operating system represents one of the most significant environmental factors in determining the operability of digital objects. Establishing an understanding of the systems that are required to interpret the information encoded within our data streams is essential to facilitate our digital curation endeavours, and GNU/Linux offers the opportunity for anyone to do so.

5.2.2 Emulation Applications for Open Source

Many information tasks that can be undertaken using proprietary tools can also be achieved with open source. If an appropriate application is not available however, the *WINE* package (Wine Is Not an Emulator) is a "Windows Compatibility Layer" for *Linux*/*Unix* which can be used to install and run many

[47] Although the entire operating system is often referred to as "Linux," Torvalds's contribution represents only a part (albeit a very significant one) of the overall system. Open source advocates tend to favour the abbreviated terminology, probably because it is more concise and catchy, which helps in its promotion.

[48] Unices is the plural of Unix.

[49] http://www.redhat.com/ [Accessed: 7 April 2005, 11:57]; http://www.SUSE.com/, [Accessed: 7 April 2005, 11:57]; http://www.mandrakesoft.com/ [Accessed: 7 April 2005, 11:57]; http://www.debian.org/ [Accessed: 7 April 2005, 11:57].

Windows applications.[50] However, the shortcomings of this project offer some insights into the problems posed when attempting to recreate proprietary, unpublished software infrastructures. Despite the WINE project's vintage[51] it still suffers from instability problems and lacks support for the full range of *Windows* applications. Legal and technological impediments associated with *Windows'* proprietary nature have been significant obstacles to the *WINE* project's success. In the event of WINE offering insufficient levels of performance or reliability, commercial *Linux* applications like *VMware* allow an alternative operating system to be installed within *Linux*, and run as an internal application.[52] This means that full software support and performance is retained, although *VMware* is not available under a free license, and any installed operating systems must also be licensed. Another proprietary alternative is to purchase **Codeweaver's** *Crossover Office* application, which builds upon *WINE* technology and offers full, robust and supported *Linux* compatibility for a very small range of *Windows* applications, including **Microsoft** *Office*, **Adobe** *Photoshop* and **Lotus** *Notes*, negating the need to purchase an additional *Windows* license.

For those wishing to run Linux or other Unix applications within a Windows environment *Cygwin* offers a "Linux-like environment for Windows", consisting of a Linux API layer and a selection of tools to provide Linux look and feel.[53]

5.2.3 Server and Development

The success enjoyed by the open source software movement can be directly attributed to a number of infrastructure, server and development technologies that have been wholeheartedly embraced by the technological community. It is in this area that open source has traditionally been most prominent, and within this domain, open source products are well established. Since most of these tools are offered at no cost and offer levels of performance, reliability and security comparable with proprietary alternatives, they appeal to many enterprises, organisations and institutions. Market share is considered in more depth in the quantitative section below.

5.2.4 The *Apache Web Server*

Alongside *GNU/Linux*, the *Apache Web Server* project represents one of the most prominent success stories of the open source movement.[54] A web server is an application used to make World Wide Web resources available. In April 2005, *Apache* had a 69% market share of all those on the Web.[55]

---

50 http://www.winehq.org/ [Accessed: 7 April 2005, 12:10].

51 The WINE project's origins can be traced to June 1993.

52 http://www.vmware.com/support/linux/ [Accessed: 7 April 2005, 12:10].

53 http://cygwin.com [Accessed: 13 July 2005, 11:49]

54 http://www.apache.org [Accessed: 7 April 2005, 12:10].

55 http://news.netcraft.com/archives/web_server_survey.

Straightforwardly configurable, well-documented, secure and available for a wide range of platforms, *Apache* pushes into second place its closest rival, **Microsoft's** *Internet Information Server*. *Apache* can be modified to suit particular deployments, allowing system administrators to customise the services they offer, effectively creating new web servers based on the *Apache* model. For digital curators the transparency offered by *Apache* is welcome, given the vast numbers of web services and web-deployed applications currently in use that are closely integrated with the web server. An understanding of these applications can only be obtained in many circumstances by understanding the software infrastructure that facilitates their delivery.

5.2.5 Databases

Databases are central to most of our interactions with digital technologies, offering storage opportunities as structured as individual applications require. Several open source packages are available. Three particular examples enjoy great prominence, with their own respective individual strengths, each offering a maturity and depth of functionality elevating them above many proprietary packages. *MySQL* is perhaps the best known, and compares extremely favourably with most proprietary equivalents, particularly in terms of speed and stability.[56] It is particularly valuable when deployed on the Web due to its quick handling of multiple connections. *PostgreSQL* and *Firebird* are other notable examples, and tend to be regarded as more functionally complete than *MySQL*.[57] None matches the heavyweight functionality offered by the leading proprietary database (*Oracle*), but unless an application has special requirements *PostgreSQL* in particular is likely to incorporate most if not all of the necessary features.[58] All three packages run natively on a range of platforms including *Linux* and *Windows*. *MySQL's* significantly larger user base accounts for its more comprehensive documentation and help structures, as well as its increased stability.[59] Prominent *MySQL* users include **Google**, **Cisco**, **Sabre Holdings**, **Hewlett Packard**, **NASA** and **Yahoo**![60]

5.2.6 The GRID

Of interest to many working in the

html [Accessed: 7 April 2005, 12:10].

[56] http://www.mysql.com [Accessed: 7 April 2005, 12:10].

[57] http://www.postgresql.org/ [Accessed: 7 April 2005, 12:10]; http://firebird.sourceforge.net/ [Accessed: 7 April 2005, 12:10].

[58] http://www.oracle.com [Accessed: 7 April 2005, 12:10].

[59] A fuller comparison of the relative merits of each can be found in "*PostGreSQL or MySQL?*", http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html [Accessed: 7 April 2005, 12:10] and Ian Gilfillan, December 2003, "*PostgreSQL vs MySQL: Which is better?*", *DatabaseJournal.com* http://www.databasejournal.com/features/postgresql/article.php/3288951 [Accessed: 7 April 2005, 12:10].

[60] http://www.mysql.com/customers [Accessed: 7 April 2005, 12:10].

scientific data disciplines is the use of GRID computing, which uses the processing power of several computers connected via a network to solve complex and large-scale computational problems. CERN[61] describes the GRID as "a service for sharing computer power and data storage capacity over the Internet.". The responsibility for defining specifications for grid computing is held by the Global Grid Forum (GGF),[62] and these are implemented through the Globus Toolkit by the Globus Alliance,[63] a group of individuals and organisations developing fundamental technologies behind the GRID. The Globus Toolkit is an open source toolkit used for building Grid systems and applications. A growing number of projects and companies are using this package to unlock the potential of grids for their own specific purposes. It has become the *de facto* standard for grid middleware and provides a standard platform for services to build upon. As they did during the development of TCP/IP, open source tools are playing a fundamental role in the development within this area, facilitating its growth and the creation of new tools and application possibilities. Similarly, open standards and collaboration are intrinsic characteristics of and fundamental requirements for the GRID.

[61] Conseil Européen pour la Recherche Nucléaire (European Laboratory for Particle Physics)

[62] http://www.gridforum.org, [Accessed: 11 July 2005, 15:43]

[63] http://www.globus.org, [Accessed: 11 July 2005, 15:48]

5.2.7 Programming Languages

Part of the *GNU/Linux* operating system, the *GNU C Compiler* (GCC) is an open source implementation of a *C* language compiler. Modules are also available to add support for a range of additional languages such as *C++*. Furthermore, a number of other languages operate under open source licenses, and several have been relied upon consistently in a range of computing areas. Three of the most prominent are the scripting languages *PHP* (PHP Hypertext Preprocessor), *Perl* (Practical Extraction and Report Language), and *Python*.[64] Because all three are traditionally interpreted languages (that is, they need not be compiled prior to execution), when it is made available their code is usually in a human-readable form. This also facilitates multi-platform interoperability. *PHP* is most widely used in the development of dynamic Web pages. Popular among Web developers due to its fast parsing and flexibility, *PHP* is also versatile and comes with many built-in and modular interfaces. Database connectivity is straightforward; while *PHP* is most commonly associated with *MySQL* it can connect to any ODBC-enabled database. *Perl* offers similar functionality to *PHP*, but with more general deployments traditionally. *Perl* is frequently used to add dynamic functionality to Web pages, but it is also used to handle a range of other tasks involved in system

[64] http://www.php.net [Accessed: 7 April 2005, 12:10]; http://www.perl.com [Accessed: 7 April 2005, 12:10]; http://www.python.org/ [Accessed: 7 April 2005, 12:10].

administration and data processing. *Perl's* rich support for regular expressions also makes it useful as a text manipulation language. Like both *Perl* and *PHP*, *Python* is frequently used in a Web environment. Combining powerful capabilities with a simple syntax, *Python* also has interfaces to numerous system calls and libraries. Additional modules can be developed using *C* or *C++*, extending the functionality to suit individual requirements. All three programming languages are portable, supporting a range of platforms including *Linux*, *Windows*, *Mac*, and *OS/2*. By developing in an open source environment, even if languages should fall into decline or obsolescence, existing code will be able to be re-purposed for a future system configuration and executed with mitigated risk of loss.[65]

5.2.8 Others

To further describe the various open source software packages that have established major footholds within the Internet infrastructure would take a great deal of space; suffice to say that OSS applications exist for almost every subject area, from eGovernment to gaming. Further examples of popular and successful tools include *Sendmail*, the world's most used email server, *Bind*, the most commonly used domain name system (DNS) server, and *Apache Jakarta Tomcat*, one of the most popular *Java* Servlet and *Java* Server Pages containers in use on the Web, providing an infrastructure for the delivery of Web-based *Java* programs.[66]

5.2.9 Desktop and Productivity

While its traditional arena of dominance since the early 1990s has been in servers and network infrastructure, the recent upsurge in popularity of open source has led to the development of a number of mature and functionally rich desktop applications that measure up well against their proprietary peers. Although notable gulfs continue to exist in some areas, several key open source desktop applications have introduced innovative and practically useful features that have been subsequently adopted into commercial proprietary applications.

5.2.10 *OpenOffice.org*

With the numerous problems associated with the proprietary and hidden nature of **Microsoft's** *Office* formats, institutions should be extremely wary of regarding *Office*-encoded data as curated. *OpenOffice* is a large-scale project, backed by **Sun Microsystems** to develop a comprehensive and transparent suite

[65] Sun Microsystems are currently involved in a public debate over the merits of opening up their currently closed-source *Java* programming language. Developments here will be well worth watching, given Java's prominence as a platform-independent, web-friendly language.

[66] http://www.sendmail.org/ [Accessed: 7 April 2005, 12:10]; http://www.isc.org/index.pl?/sw/Bind/ [Accessed: 7 April 2005, 12:10]; http://jakarta.apache.org/tomcat/ [Accessed: 7 April 2005, 12:10].

of tools, including word processor, spreadsheet and presentation software, that writes to open standard formats defined in XML.[67] With its current version (1.1.4), the project has enjoyed success. 'Read' and 'write' support for **Microsoft** *Office* formats is included, but since these are not publicly documented, errors are occasionally encountered, particularly when dealing with complex structures such as tables. Objects such as images, plugins, videos and charts can be embedded as straightforwardly as with **Microsoft** tools. There is no support for *Visual Basic for Applications* (VBA) macros, since this is a proprietary **Microsoft** technology, but scripting is possible using the integrated *StarBasic* syntax. The imminent new release of *OpenOffice* will include support for the OASIS[68] OpenDocument format, which is likely to be adopted by the European Commission as the recommended format for document interchange within the European public sector.

An **eWeek** survey comparing *OpenOffice 1.1* and **Microsoft's** *Office 2003* illustrates the relative merits of the two application suites.[69] The general consensus is that while *OpenOffice* represents a good free package, with several unique features such as built-in PDF-writing support and a user interface that integrates each of the individual applications, it lacks the polish and some of the more advanced functionality of the latest version of *Office*. *OpenOffice 1.1* is regarded as functionally comparable to **Microsoft's** *Office 97*, although the open source product is thought to offer several additional features and a greater level of reliability. Many ordinary users are unlikely to have requirements that extend beyond the features that are included. However, when Jack Wallen Jr. of **ZDNet** Australia writes "if you can do it in **Microsoft** *Office*, you can do it in *OpenOffice.org…* for free," it should be borne in mind that *OpenOffice* still has some way to go before matching all of the functionality of **Microsoft**'s flagship application.[70] That it exceeds **Microsoft**'s efforts in terms of implementing a system for the creation, editing and rendering of preservable documents is however, unquestionable.

5.2.11 The Mozilla Project

Comprising a number of individual programs, the *Mozilla* project represents one of the most successful open source desktop application projects, offering a level of maturity, functionality and innovation that matches and surpasses much equivalent proprietary software.[71] At its forefront is the *Mozilla*

[67] http://www.openoffice.org [Accessed: 7 April 2005, 12:13].

[68] Organisation for the Advancement of Structured Information Standards, http://www.oasis-open.org/home/index.php , [Accessed: 7 July 2005, 14:39]

[69] Jason Brooks, April 2004, "*Office 2003 vs. Openoffice.org*", *Eweek.com*, http://www.eweek.com/article2/0,1759,1571626,00.asp [Accessed: 7 April 2005, 12:10].

[70] http://www.zdnet.com.au/insight/0,39023731,20270300,00.htm [Accessed: 7 April 2005, 12:10].

[71] http://www.mozilla.org/ [Accessed: 7 April 2005, 12:10].

package, which offers Web browsing, email, newsgroup and IRC access and a Web development application within a single, customisable interface. It is particularly in the area of security that *Mozilla* outshines **Microsoft's** frequently vulnerable *Internet Explorer* and *Outlook Express*, but it also offers greater compliance to web standards and improved functionality, with a built-in in pop-up blocker and integrated search facility that unless patched with **Microsoft**'s *Windows XP Service Pack 2*, *Internet Explorer* does not offer.

The *Mozilla* project has also developed a range of sister products. *Mozilla Firefox* is a stripped down, lightweight Web browser with support for the addition of separate modules of functionality, or extensions.[72] It aims to be fully customisable, with optional features that introduce exciting navigational and development possibilities. *Thunderbird* is another project, essentially promising the same things for email as *Firefox* offers for the Web.[73] Lightweight and secure, it supports all major protocols and can be fully customised to suit environment or user preferences. All of these *Mozilla* tools are available for *Windows*, *Linux*, and *Mac OS X*, emphasising the interoperability of these solutions.

The *Mozilla* project has inspired and enabled the development of a number of other applications, including the defect-tracking system *Bugzilla*, which facilitates the reporting of errors from a wide number of applications, ensuring their continued development and refinement.[74]

5.2.12 Specific Open Source Applications for Digital Curation

The increasing maturity of open source software has led to the development of a range of tools designed for the achievement of specific, specialist goals. A number of factors make the open source model particularly suitable for the digital curation community, and this has led to concentrated development in this area. The requirements for openness and 'future-proofing' within software and the often physically distributed nature of organisations and projects are issues in which open source can be profoundly beneficial.

The following sections detail a number of prominent open source applications aimed specifically and indirectly at meeting digital curation requirements.

(a) Fedora Digital Object Repository Management System

The *Fedora* project (not to be confused with **Red Hat**'s *Fedora Linux* distribution), originally developed by the Digital Library Research Group at Cornell University is one of several digital object repository architectures that have been proposed in recent years. The *Fedora* structure is based on object models that

[72] http://www.mozilla.org/products/firefox/ [Accessed: 7 April 2005, 12:10].

[73] http://www.mozilla.org/projects/thunderbird/ [Accessed: 7 April 2005, 12:15].

[74] http://bugzilla.mozilla.org/ [Accessed: 7 April 2005, 12:15].

each form the template for individual units of content, called data objects. These can contain various digital content, associated metadata and references to representation information. Behaviour objects describe tools and services that can be used by the repository to provide access to the data objects. The system has a three-layered architecture. The Web Services Exposure layer defines interfaces for administration and access, the Core Subsystem layer implements their subsystems and the Storage Layer implements the storage subsystem that handles reading, writing, and removal of data from the repository. Digital Objects are stored as XML files corresponding to an extension of the Metadata Encoding and Transmission Standard (METS). Among *Fedora*'s "noteworthy" features identified by D-Lib Magazine[75] are its XML submission and storage, its access control and authentication, searching, OAI-PMH[76] Metadata harvesting and a batch utility supporting the mass creation and loading of data objects. A recent paper entitled "*Fedora*: An Architecture for Complex Objects and Their Relationships"[77] describes further notable qualities, in particular the fact that the software is implemented as a set of Web Services and that its full functionality is exposed through a series of well defined web service APIs. The D-Lib article describes four use case scenarios for *Fedora*, illustrating its usefulness. From the first "low barrier to entry" scenario to a full fledged digital library or repository for distributed objects *Fedora* is sufficiently flexible to meet the requirements of many institutions. Undoubtedly innovative, and functionally rich, *Fedora* offers a good illustration of what open source development can achieve.

(b) DSpace

The *DSpace* Institutional Repository System, developed jointly by MIT and **Hewlett Packard** offers capture, storage, indexing, preservation and redistribution functionality. It aims to satisfy the definition of an Institutional Repository offered by Clifford Lynch, as "an organisational commitment to the stewardship of digital materials, including long-term preservation where appropriate, as well as organisation and access or distribution".[78]

---

[75] Thornton Staples, April 2003, "*The Fedora Project: An Open-source Digital Object Repository Management System*", *D-Lib Magazine*, Volume 9, Number 4, http://www.dlib.org/dlib/april03/staples/04staples.html [Accessed: 7 April 2005, 12:15].

[76] Open Archive Initiative Protocol for Metadata Harvesting, http://www.openarchives.org/OAI/openarchivesprotocol.html [Accessed: 7 July 2005, 14:53]

[77] Carl Lagoze, Sandy Payette, Edwin Shin, Chris Wilper, (rev v.4, March 2005), "*Fedora: An Architecture for Complex Objects and Their Relationships"* , http://www.arxiv.org/pdf/cs.DL/0501012 [Accessed: 7 April 2005, 12:15].

[78] Clifford A. Lynch, February 2003,"*Institutional Repositories: Essential Infrastructure for Scholarship in the Digital Age*" *ARL*, no. 226: 1-7, http://www.arl.org/newsltr/226/ir.html [Accessed: 7 April 2005, 12:15].

DSpace can be configured to accept a diversity of digital content ranging from documents, books and theses to data sets, computer programs, visual simulations and models. These are organised according to the groups that contribute content, called 'communities', and 'collections', which house individual content items and files. The primary goal of *DSpace* is digital preservation, to provide long-term physical storage and management of materials in a secure and effectively administered environment. Persistent identifiers are allocated to each stored item in the interests of ensuring their longevity, with preservation conducted in both bit and functional terms. The latter is achieved using emulation or migration strategies for supported open formats and by relying on the third party tools that are expected to emerge for popular proprietary formats. It is conceded that for unknown or one-off proprietary data functional preservation is difficult, but by preserving the bit-stream too it is hoped that future digital archaeologists will at least have the opportunity to retrieve and reproduce information. Ancillary *DSpace* functionality allows the implementation of access controls, versioning and search and retrieval based on Dublin Core metadata which can be applied to each submitted object. *DSpace* is promoted as a flexible solution, equipped to effectively handle the diversity of materials and expectations implicit within a multi-disciplinary archive, mainly through its use of communities in the organisation of its information. Built-in *Java* APIs allow the interoperation of stored content with other systems that an institution may maintain.

(c) FreeBXML

ebXML (Electronic Business using eXtensible Markup Language)[79] is a suite of specifications that facilitate the exchange of business information over the Internet by organisations of any size. Using these specifications it is possible for organisations to exchange messages, trade, communicate in common terminology and define and register processes relevant to their business. Started in 1999 by OASIS and the UN/ECE agency CEFACT it was based upon five layers of substantive data specification, which are realised in XML standards for business processes, core data components, collaboration protocol agreements, messaging and registries and repositories.[80] *freebXML* is an initiative aiming to promote the *ebXML* specifications through the sharing of software, expertise and experience. Its web site (http://www.freebxml.org) offers centralised access to relevant code and applications as well as a forum for discussion about developments and deployments using *ebXML*. Among the most relevant programs available from the web site under open source licenses are *freebXML*

[79] http://www.ebxml.org/.

[80] For further information see Brian Gibb, Suresh Damodaran, 2002, "*ebXML : Concepts and Application*", Wiley, ISBN**:** 076454960X, or Alan Kotok and David R.R. Webber, 2001, "*ebXML: The New Global Standard*", New Riders, ISBN: 0735711178.

CC, a set of tools developed to facilitate the management of data dictionaries and *freebXML Registry*, built upon an extensible information model, allowing the storage of any kinds of data in its repository, with arbitrary associations created between registry entries.

(d) JHOVE

The result of collaboration between JSTOR[81] and Harvard University Library, the JSTOR/Harvard Object Validation Environment[82] aims to provide functions to perform format-specific identification, validation and characterisation of digital objects. In essence, this offers the opportunity to automatically determine a particular object's format if unknown, to confirm or deny the validity of a purported example of a particular format by assessing whether it meets syntactic and semantic requirements of that format and to identify the intrinsic properties of a particular object based on its format. Standard format modules are distributed with the system and include AIFF, ASCII, GIF, HTML, JPEG, PDF, TIFF and XML.

(e) LOCKSS

LOCKSS stands for "Lots of Copies Keeps Stuff Safe"[83] and is an open source peer-to-peer application with its core *raison d'etre* to offer persistent access to preserved digital materials. Initiated by Stanford University Libraries, *LOCKSS* runs on standard desktop workstations and offers librarians and information managers the opportunity to create low-cost, persistent and accessible copies of digital content as it is published. A secure peer-to-peer polling and reputation system ensures that the integrity and accuracy of *LOCKSS* materials are maintained.

(f) Xena

The National Archives of Australia originally developed the *XML Electronic Normalising of Archives*[84] project to meet the challenges posed by preserving electronic records into the future in a constantly changing hardware and software culture. The application aims to resolve these concerns by converting electronic records in proprietary formats to a standardised XML format that can be read by future technology. *Xena*'s current version supports a range of formats that can be converted with no information loss to the standard XML. These include **Microsoft**'s *Word*, *Excel* and *Powerpoint*, the *OpenOffice.org* suite of formats, RTF, Relational database files, JPG, GIF, TIFF, PNG and BMP image files, HTML and plain text. Furthermore, with its plug-in based architecture, *Xena* can conceivably be extended to support any other formats.[85]

---

[81] The Scholarly Journal Archive, http://www.jstor.org [Accessed: 25 July 2005, 11:43]

[82] http://hul.harvard.edu/jhove/ [Accessed: 25 July 2005, 11:43]

[83] http://lockss.stanford.edu/ [Accessed: 25 July 2005, 11:43]

[84] http://xena.sourceforge.net/ [Accessed: 25 July 2005, 11:43]

[85] For an example of an open source tool which 'normalises' file formats for preservation see Adam

5.2.13 Other Institutional Repository Implementations

Developed at the University of Southampton, *GNU EPrints* facilitates the creation of online archives, with the default configuration a repository of the research output of an academic institution. *EPrints* servers are designed to help dissemination of research publications by sharing associated metadata using Open Archive Initiative (OAI) standards. Further open source alternatives include *MyCoRe*, developed in Germany, the Dutch ARNO and also CERN Document Server Software. According to a recent DPC Technology Watch Report by Paul Wheatley into "Institutional Repositories in the Context of Digital Preservation"[86] none of these four solutions cites digital preservation as a key aim.

With the range of repository solutions available there is a increasing interest into how multiple repositories might cooperate within a global curation network. The transparent nature of open source facilitates the implementation of systemic connections offering a range of potential benefits. Cooperation on the selection of content and optimisation of technical infrastructures can take place, and duplication of effort can be minimised.

Open source carries other advantages in the context of digital resource registries and repositories. For instance, Representation information registries can benefit from its flexible legal status by offering direct access to rendering, management and conversion applications that are distributed under open source licensing agreements.

Rusbridge, April 2004, "*XENA: Electronic Normalising Tool*", *DigiCULT.Info*, Issue 7, page 32, http://www.digicult.info/pages/newsletter.php [Accessed: 7 April 2005, 11:49].

[86] Paul Wheatley, 2004, "*Institutional Repositories in the context of Digital Preservation*" *DPC Technology Watch Series* Report 04-02.

6 Quantitative Issues

6.1 Financial Costs of Open Source Software

It is important to consider financial implications in terms of total cost of ownership (TCO), particularly when software is distributed free of charge. However, since a definitive list of the cost factors that should be taken into account in a TCO study has yet to be settled upon, a number of conflicting accounts exist. It is no doubt possible to identify a persuasive TCO study in favour of most software configurations, but the actual figures will always depend on a specific combination of environment and requirements. An accurate picture can only be drawn following consideration of all the relevant individual cost elements, from software and hardware purchases to administration, and from technical support to staff training.

As well as ambiguities across surveys it is also clear that no one has yet conducted any formal studies as to the relative cost implications of conducting a digital curation strategy with open source or proprietary tools. This is true both when considering the relative costs of curating digital resources that are open source or proprietary and when considering using open source tools to conduct our digital curation activities. While it seems likely that the use of open tools and software formats are likely to prolong the longevity of our digital assets it is hard to present the cost implications of this hypothesis in quantitative terms. Instead we must rely on the figures that do exist which describe the relative cost implications of using open source and proprietary tools in more general ways. Inevitably there will be unique implications relating to the cost of digital curation, but these are difficult to assess at this time.

6.2 Software Acquisition and Upgrade Costs

The initial acquisition cost of open source software will usually be less than any proprietary alternative. Of course, it need not be free (i.e. gratis) under the terms of its license, and other additional costs may be incurred for documentation, storage media, and support contracts. Taking these factors into account, a 2001 study by **Cybersource Consulting** found the following acquisition cost results, illustrating the scalability of an open source solution over three increasingly sized installation environments:[87]

[87] Cybersource, 2004, "*Linux vs. Windows Total Cost of Ownership Comparison*", http://www.cyber.com.au/cyber/about/linux_vs_windows_tco_comparison.pdf [Accessed: 7 April 2005, 12:18].

	Microsoft	*GNU/Linux*	OS Savings
50 users	$69,987 (€56,615)	$80 (€65)	$69,907 (€56,550)
100 users	$136,734 (€110,609)	$80 (€65)	$136,654 (€110,544)
250 users	$282,974 (€228,907)	$80 (€65)	$282,894 (€228,842)

The reason why open source software scales so well is that it is only necessary to purchase or obtain a single license, which covers unlimited subsequent installations. Proprietary software, on the other hand, is typically licensed on a per-installation or per-user basis. This is worth bearing in mind if the intention is to deploy a large number of workstations. Network World Fusion News reported in 2001 that a major part of the reason for an increase in *Linux's* deployment in finance, healthcare, banking and retail was its scalability in cost and technical terms when large numbers of identical sites and servers are needed. The journal calculated that for a 2,000-site deployment **SCO** *UnixWare* would cost $9 million (€7.3m), *Windows* $8m (€6.5m), and **Red Hat** *Linux* just $180 (€146).[88]

Upgrade costs also compare favourably using open source applications. Proprietary upgrades will typically cost around half the amount of the original application. Users can subsequently find themselves at the mercy of the proprietary companies, who have a monopoly on the distribution of their software. To upgrade an open source application one simply has to download the latest version, or pay the original cost once again and redeploy across as many machines as required.

6.3 License Management and Litigation Costs

Needless to say, open source software is again favourable in the context of license management and litigation. For users of proprietary software, failure to adhere to the strict terms of software licenses can lead to extremely heavy fines and even custodial sentences. It is therefore in users interests to manage licenses effectively, undertaking regular software audits and even installing license-tracking software. Under an open source license such procedures, costly in terms of both time and money, are rendered unnecessary. Similarly, the costs involved in migrating to and from open source formats, emulation of existing open source software architectures and of supplying open source tools to render or convert particular file formats is likely to be dramatically less than with proprietary alternatives.

6.4 Hardware Costs

As far as hardware is concerned, it is generally acknowledged that open source

[88] Deni Connor, March 2001, "*Linux Slips Slowly into the Enterprise Realm*", http://www.nwfusion.com/news/2001/0319specialfocus.html [Accessed: 7 April 2005, 12:15].

software – such as the *GNU/Linux* operating system – can run effectively on a lower specification machine than its *Windows* equivalent. The latest version of *Windows*, *XP Professional*, recommends a 300Mhz Intel compatible processor, 128MB of RAM and a minimum of 1.5GB of hard disk space. A comparable version of **Mandrake** *Linux* (version 9.1) requires only a Pentium class processor, 128MB of RAM and 150MB of disk space. As *Linux* desktops and user interfaces have become more graphically complex the margin between the two has narrowed. But since a *Linux* system tends to be much more configurable, it is more straightforward to install only those parts of a system that are really required, saving processor power and disk space.[89] Furthermore, unlike existing consumer *Windows* systems most *Linux* distributions are available in both 32bit and 64bit **Intel**-based versions, offering users the opportunity to exploit the advantages of newer more sophisticated hardware. Further emphasising the interoperability intrinsic to open source, many *GNU/Linux* distributions are available for non-**Intel** hardware, such as *PowerPC*, **Sun** *SPARC* and *Alpha*. The transparency at *Linux*'s core facilitates porting to limitless alternative hardware environments, and numerous endeavours are constantly being undertaken to run Linux on devices as varied as **Apple**'s *iPod* digital music device[90] and **Microsoft**'s *X-Box* gaming console.[91] With the uncertainty surrounding tomorrow's computer hardware environments it is comforting to know that such migration remains possible, irrespective of the nature of hardware products and their original intended purposes. Suffice to say, from a cost perspective the flexibility of *Linux* mitigates the likelihood of hardware obsolescence, offering users more opportunity to make the most of their existing resources. Furthermore, rumours suggest that future versions of **Microsoft** *Windows* will incorporate hardware-tied security, essentially limiting the user's ability to configure the hardware environment on which the software operates.[92] This may have problematic consequences for future migration and reuse.

6.5 Support and Training

For other, less explicit up-front costs it becomes more difficult to find consensus. Technical support and administration is one such area. **Microsoft** claims that it is more straightforward to find trained administrators and technicians for its platforms, and that they therefore cost less. However, the open source community rebuts this, arguing that with

[89] This also means that hardware can be used for longer, without the need to upgrade so frequently.

[90] http://neuron.com/~jason/ipod.html [Accessed: 7 April 2005, 14:35].

[91] http://www.xbox-linux.org/ [Accessed: 7 April 2005, 14:57].

[92] http://www.theregister.com/2003/11/03/ms_to_intro_hardwarelinked_security/ [Accessed: 25 July 2005, 11:55]

GNU/Linux fewer administrators are required, because it is possible to automate a great deal and the systems are more reliable. Support for open source products may be less available in a formal commercial capacity, but many problems encountered in one's digital curation efforts can be mitigated by turning to a vast web-based community of users and experts that has demonstrated a regular and enthusiastic commitment to offer assistance.

A further question is that of training. Anecdotal evidence suggests that the costs involved are fairly modest, thanks to the proliferation of modern GUI desktops within *Linux* systems. It remains to be seen whether this is demonstrably true across the board, although it could be argued that training costs should be no more than those incurred for *Windows* training. However, retraining experienced *Windows* users will inevitably be more challenging, and will involve higher associated costs.

6.6 Total Cost of Ownership

The Robert Frances Group's July 2002 study found that the TCO of *GNU/Linux* is roughly 40% of that of *Windows*, and 14% of **Sun Microsystems'** *Solaris*.[93] The group used actual costs of production deployments of web servers at fourteen Global 2000 enterprises, basing its analysis on software, and hardware purchases and maintenance, upgrade and administrative costs. This study also found that although *Windows* administrators cost less individually, each *Linux* or *Solaris* administrator could cover many more machines, making *Windows* administration more expensive. It was also revealed that *Windows* administrators spent twice as much time patching systems and dealing with security issues than the others.

There is also a great deal of persuasive testimonial evidence from a range of companies and public institutions that have used open source successfully and saved money. For instance, **Amazon.com** was able to cut US$17m (€13.8m) in technology expenses in a single quarter by moving to *Linux*. The city of Largo in Florida saved $1m (€811,000) by using *GNU/Linux* and 'thin clients,' and **Intel** Vice President Doug Busch reported savings of $200m (€162.4m) by replacing proprietary *UNIX* servers with *GNU/Linux* alternatives.[94] While TCO studies are useful for interest

[93] David A. Wheeler, 2005,"*Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*", http://www.dwheeler.com/oss_fs_why.html [Accessed: 7 April 2005, 11:30].

[94] Lynn Haber, April 2002, "*City Saves with Linux, Thin Clients*", ZDNet, http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860180,00.html [Accessed: 7 April 2005, 12:15]; David A. Wheeler, 2005,"*Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*", http://www.dwheeler.com/oss_fs_why.html [Accessed: 7 April 2005, 11:30]; Stephen Shankland, Margaret Kane and Robert Lemos, October 2001, "*How Linux Saved Amazon Millions*", *News.com*, http://news.com.com/2100-1001-275155.html?legacy=cnet&tag=owv [Accessed: 7 April 2005, 12:15].

purposes, the source of their commissioning should be carefully noted. At least one (albeit **Microsoft** sponsored) study has suggested that *Windows* is cheaper than *Linux*,[95] although technology writer Joe Barr has discussed and criticised some of the problems inherent in the report, such as assuming no upgrades over a five year period, costing for an older operating system, and not using the current **Microsoft** Enterprise license. Barr concludes his report by stating that "TCO is like fine wine: it doesn't travel well. What may be true in one situation is reversed in another. What gets trumpeted as a universal truth ('*Windows* is cheaper than *Linux*') may or may not be true in a specific case, but it is most certainly false when claimed universally."[96] Conversely, it is very unlikely that for every configuration Linux represents a more cost-effective solution.

6.7 Longer-term Considerations

Of particular interest to the digital curator are the long-term cost implications of using open source software. As suggested above, there are few (if any) conclusive sources offering quantitative savings information, but the reusability and accessibility implicit in open source will result in inevitable cost savings when long-term access to digital materials is required. The costs of recovery of information are becoming increasingly well known. Several insightful examples come from the US legal system, where information discovery legislation compels litigants to produce electronic materials prior to trial, at the defendant's expense.[97] Proprietary systems can cause problems in the face of these kinds of requirements, hampering straightforward access to digital materials and bottlenecking the legal process. In the case of *Zubulake v. UBS Warburg,*[98] the retrieval and presentation of content from a single email stored on backup tapes was priced at $175,000. In another similar example, in the case of *Murphy Oil Corporation v Fluor Daniel*[99] it was stated that that the recovery of email content into a presentable form would cost some $6.2 million dollars and take more than six months, excluding attorney time. UK Freedom of Information Legislation creates similar obligations to provide information, which once again can be problematic within a proprietary or

[95] IDC, "*Windows 2000 Versus Linux in Enterprise Computing*", http://www.microsoft.com/windows2000/docs/TCO.pdf [Accessed: 7 April 2005, 12:15].

[96] David A. Wheeler, 2005,"*Why Open Source Software/Free Software (OSS/FS, FLOSS, or FOSS)? Look at the Numbers!*", http://www.dwheeler.com/oss_fs_why.html [Accessed: 7 April 2005, 11:30].

[97] National Electronic Commerce Coordinating Council, 2004, "*Effectively Managing the Discovery of Electronic Records*", http://www.ec3.org/Downloads/2004/Effectively_Man_Discovery_of_El_Records.pdf [Accessed: 7 April 2005, 12:15].

[98] Zubulake v. UBS Warburg LLC 217 F.R.D. 309 (S.D.N.Y. 2003) [Note: Zubulake I, Opinion of 13 May 2003], Zubulake v. UBS Warburg LLC, 216 F.R.D. 280 (S.D.N.Y. 2003) [Note: Zubulake II, Opinion of 24 July 2003]

[99] Murphy Oil v. Fluor Daniel, Inc., 2002 WL 246439 (E.D. La. 19 Feb 2002)

opaque system. These sums represent the cost of inaccessible digital resources. By introducing transparency at every level of one's information infrastructure, in terms of both technology and comprehension, such costs can inevitably be mitigated. Irrespective of legal obligations to present information, long-term use inevitably requires the repackaging and migration of digital materials to accommodate the technological environments and standards of the day. Only by establishing and effectively documenting an understanding of our digital resources can such activities be more straightforwardly - and more cost-effectively - undertaken.

6.8 Performance and Reliability

Performance and reliability are important factors in determining the contemporary and long-term usefulness of our digital assets, and further empirical evidence suggests that open source technologies compare favourably in these terms with their proprietary peers. Again, there are few formal quantitative studies based explicitly around the performance of digital curation applications or curation processes, but one may regard the more generic examples that do exist as a useful barometer. According to a study undertaken at the University of Wisconsin in 2000, 21% of *Windows 2000* applications crashed when presented with random testing using valid keyboard and mouse input.[100] An additional 24% of applications hung when presented with valid keyboard and mouse input. When the same test was undertaken five years earlier using a then current *Linux* distribution, the failure rate was just 9%; and since then the reliability of open source software has improved. Comparable studies by **IBM** and **Bloor Research** have had similar results.[101]

An eWeek survey in 2002 found that *MySQL* was comparable to the proprietary market leader, *Oracle*, and offered better performance than a number of other proprietary applications, including **Sybase Inc's** *ASE*, **IBM's** *DB2* and **Microsoft's** *SQL Server 2000 Enterprise Edition*.[102]

As far as performance is concerned, the results for open source are also promising. As with TCO, performance benchmarks are often dependent on environment, as well as whatever assumptions the tester has made; the only real benchmark that can be of value to an individual user is the one that most closely mirrors the work actually being done. PC Magazine found in November 2001 that *Linux* with *SAMBA* significantly outperformed *Windows 2000*. At

---

[100] Justin E. Forrester and Barton P. Miller, 2000, "*An Empirical Study of the Robustness of Windows NT Applications Using Random Testing*", ftp://ftp.cs.wisc.edu/paradyn/technical_papers/fuzz-nt.pdf [Accessed: 7 April 2005, 12:15].

[101] Li Ge, Linda Scott and Mark VanderWiele, 2003, "*Putting Linux Reliability to the Test*", http://www-106.ibm.com/developerworks/linux/library/l-rel/ [Accessed: 7 April 2005], 12:15; http://gnet.dhs.org/stories/bloor.php3 [Accessed: 7 April 2005, 12:15].

[102] Timothy Dyck, February 2002, "*Server Databases Clash*", *eWeek*, http://www.eweek.com/article2/0,3959,293,00.asp [Accessed: 7 April 2005, 12:15].

one stage in the test, using a 1Ghz Pentium 3 with 512MB of RAM and handling thirty client connections, the *Linux* software was 78% faster than **Microsoft's**.[103] In February 2003, a team of physicists broke the Internet2 Land Speed Record using *GNU/Linux*, sending 6.7GB of uncompressed data from Sunnyvale, California to Amsterdam in the Netherlands in just 58 seconds.[104]

6.9 Market Share

The market share enjoyed by open source software is significant from a digital curation perspective since it is likely to determine to a great extent the demand from within the community for preservation of these digital assets and their associated information. While not the only important factor, it is often advantageous in the interests of longevity to go with what most people are using. Generally speaking, it is marginalised or minimally adopted hardware, software and standards that are more likely to become irretrievably lost. Several examples exist where open source software leads the field, or enjoys prominence on a commensurate level with its proprietary equivalents.

Serving Web pages is one of several areas where open source software is dominant. According to **Netcraft's** statistics on web servers the *Apache Web Server* was responsible for some 70% of all Web pages in July 2005, with the closest rival **Microsoft's** *Internet Information Server* responsible for just 23%. *GNU/Linux* is the second most prevalent operating system for web servers with 29%, behind *Windows* which has just under half of the entire market share. Other open source operating systems (such as FreeBSD) comprise around 6% of all those that serve Web pages.[105]

Open source software enjoys prominence in other areas of the Internet too. *Sendmail* is the leading email server, with 42% of the market share.[106] The DNS server *Bind*, an application that translates human-readable Web site names into a format understandable by computers, had a 95% market share in 2000.[107] Further emphasising the web-based prominence of open source, *PHP* is the most commonly used Web programming language in the world, running on over eighteen million sites during January 2005, outstripping its primary rivals *ASP.NET*, *Java Server Pages*, and *Cold Fusion*.[108]

---

[103] Oliver Kaven, November 2001, "*Performance Tests: File Server Throughput and Response Times*", Pcmag.com, http://www.pcmag.com/article2/0,1759,16227,00.asp [Accessed: 7 April 2005, 12:15].

[104] Katie Dean, February 2003, "*Data Flood Feeds Need for Speed*", *Wired*, http://www.wired.com/news/infostructure/0,1377,57625,00.html [Accessed: 7 April 2005, 12:15].

[105] http://news.netcraft.com/, [Accessed: 7 April 2005, 12:25]

[106] http://cr.yp.to/surveys/smtpsoftware6.txt, [Accessed: 7 April 2005, 12:25]

[107] Bill Manning, "*in-addr version distribution*", http://www.isi.edu/~bmanning/in-addr-versions.html [Accessed: 7 April 2005, 12:25].

[108] http://www.php.net/usage.php, [Accessed: 7 April 2005, 12:25]

At a more general level, part of the findings of the Free/Libre and open source Software (FLOSS): Survey and Study published in June 2002 found that 43.7% of German establishments, 31.5% of British establishments, and 17.7% of Swedish establishments reported using open source or free software.[109] Open source is well established within the infrastructure of the Internet, but it is only in relatively recent times that appropriate software has become available to make *Linux* a viable choice for desktop computer users. Improvements in graphical user interfaces (GUIs) and the potential to use *Linux* without recourse to unfamiliar command line instructions have made it a more appealing prospect for casual or non-expert users. A number of companies and organisations are planning or beginning migration. Public sector institutions, who require autonomy over their software infrastructures and digital preservation straightforwardness are understandably among the most enthusiastic.

[109] http://www.infonomics.nl/FLOSS/, [Accessed: 7 April 2005, 12:25]

7 Future Developments

It seems likely that the future of open source is assured, as projects continue to gather momentum, established applications mature and more and more specialist tools become available. There is a growing realisation within the software development world that openness is a desirable quality and that relying too much on the commercially driven solutions distributed by proprietary developers may threaten the longevity of one's digital assets. An open source software infrastructure, dealing in open and standardised formats ensures that transparency is maintained from the conception of information until its long-term storage. Of course, a range of factors, both financial and behavioural mean that the digital realm is unlikely to be homogenised in the near future, and it is certain that digital curators will have to continue to find imaginative ways to successfully manipulate, migrate and re-use digital information that can't be so straightforwardly comprehended. While in principle the adoption of open standards is of great value to all kinds of organisations it is unrealistic to expect all important business decisions to be made based solely on archival considerations. The open source community must continue to develop solutions that offer immediate benefits in every sense, not just in terms of digital curation, but for all levels of digital creators and users, from IT administrators, managers, research scientists, and digital librarians to students, teachers and home computer enthusiasts. Continued innovation, constant refinement and an emphasis on the financial savings and legal freedom associated with open source will help to convince users of the intrinsic benefits. It is unlikely that any battle will be won simply by emphasising the superior archival characteristics of open source software and digital assets encoded with open formats. In many cases it will require even more than just functional superiority to convince users of the benefits of open source, due to the high impact marketing that accompanies and champions many commercial products. A consumer-oriented example is in the case of portable digital audio. Despite many experts agreeing that the open *Ogg Vorbis* format offers a better encoding algorithm in terms of sound quality per byte, most users are still encoding millions of hours of their music collections in the proprietary and patented MP3 format, buoyed in their endeavours by television and web publicity, to the extent that now "digital audio" is almost synonymous in the popular consciousness with MP3. Only by continuing to offer innovative, usable and functionally rich solutions can open source and its inherent digital curation qualities expect to be fully exploited.

8 Conclusion

The open source and free software development and distribution philosophies are now well established, and offer several benefits throughout digital curation work flow. The conceptual key to understanding the value of these technologies and their associated standards is to understand the significance of transparency. By understanding the precise nature of our own digital assets and those we seek to integrate with and exploit elsewhere we are able to take more effective steps towards ensuring their continued and long-term accessibility. Removing restrictive legal barriers to this endeavour facilitates the digital curation process further still. Open source solutions arrive free from commercially motivated opacity and represent a consensus in favour of continued accessibility, comprehension and reusability. By their nature they facilitate integration and interfacing with existing infrastructures. Standardisation of formats is an unprofitable concept for those within a competitive market economy who are mainly interested in promoting their own unique product path. Unfortunately, it is also a great *facilitator* for straightforward long-term digital curation, its promotion and presence rendering the process significantly less irksome. Open source technology is founded on an unwillingness to reinvent or monopolise the wheel; a sentiment that through active collaboration our software and digital assets can be more effectively structured, injected with greater functionality and made more sustainable over the long-term. There is money to be made through open source software, but it is a consequential thing, and seldom do economic concerns drive or direct the development and release agenda. By embracing open source tools where they are available and functionally sufficient, our digital materials will be more easily comprehensible in the future. Many creators, custodians and re-users of digital information have a responsibility to ensure that the materials they create offer maximum usability and accessibility in the contemporary and are guarded against the problems of obsolescence in the future. Using open source software and open standards as a facilitator to this is a worthwhile starting point, and combined with other digital curation strategies can be effective.

Bibliography

Print

Dibona, C., S. Ockman, M. Stone, 1999, Open Source*s: Voices from the Open Source Revolution*, Sebastopol, CA, O Reilly

Dubois, P. *MySQL*, 2003, Indiana: Sams Publishing,

Gibb, B. S. Damodaran, 2002, *ebXML : Concepts and Application*, Wiley, ISBN: 076454960X

Kotok, A., D.R.R. Webber, 2001, *ebXML: The New Global Standard*, New Riders, ISBN: 0735711178

Moody, G., 2002, *Rebel Code: Linux and the Open Source Revolution*, Penguin

Netproject, 2003, *The IDA Open Source Migration Guidelines*, European Communities

Raymond E. S., 2001, *The Cathedral and the Bazaar - Musings on Linux and Open Source by an Accidental Revolutionary (Revised edition)*, Sebastopol, CA, O Reilly

Ross, S., A. Gow, 1999, *Digital archaeology? Rescuing Neglected or Damaged Data Resources*, London & Bristol: British Library and Joint Information Systems Committee, ISBN 1900508516

Ross, S., M. Donnelly, M. Dobreva, D. Abbott, A. McHugh, A. Rusbridge, 2005, *Digicult Technology Watch Report 3*

Stallman, R. M, J. Gay, L. Lessig, 2002, *Free Software, Free Society Selected Essays of Richard M. Stallman*, Free Software Foundation

Stanescu A, 2005, *Assessing the durability of formats in a digital preservation environment: The INFORM methodology*, OCLC Systems and Services, International Digital Library Perspectives, Vol 21, Number 1, 2005, pp 61-81

Weber, S., 2004, *The Success of Open Source*, Harvard, MA, Harvard University Press

Welsh, M, M. K. Dalheimer, T. Dawson, L. Kaufman, 2003, *Running Linux*, Sebastopol CA, O Reilly

Williams, S., 2002, *Free as in Freedom Richard Stallman's Crusade for Free Software*, Sebastopol, CA: O Reilly

Online

Abbott, D., *Overcoming the Dangers of Technological Obsolescence: Rescuing the BBC Domesday Project*, *DigiCULT.Info 4*, Page 4 , http://www.digicult.info/pages/newsletter.php [Accessed 7 April 2005, 11:49].

Berlind, D., 30 July 2002, *Who Gave Microsoft Control of your IT Costs? You Did, ZDNET.com,* http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2875958,00.html[Accessed: 7 April 2005, 12:30]

Bernstein D. J., 2001, *Internet Host SMTP Server Survey* http://cr.yp.to/surveys/smtpsoftware6.txt

[Accessed: 7 April 2005, 12:30]

Bishop,T., 12 September 2003, *Should Microsoft Be Liable For Bugs?,* Seattle Post Online, http://seattlepi.nwsource.com/business/139286_msftliability12.html [Accessed: 7 April 2005, 12:30]

Books from the Past http://www.booksfromthepast.org [Accessed: 7 April 2005, 13:31]

Brooks, J, 26 April 2004, *Office 2003 vs. OpenOffice.org*, *eWeek* http://www.eweek.com/article2/0,1759,1571626,00.asp [Accessed: 7 April 2005, 13:31]

Cameron, N., September 2003, Open Source *Bookmarks Australian Heritage*, http://www.computerworld.com.au/index.php?id=522130461&fp=16&fpid=0 [Accessed 7 April 2005, 11:56].

CAMiLEON BBC Domesday Rescue Project: http://www.si.umich.edu/CAMILEON/domesday/domesday.html [Accessed: 7 April 2005, 13:31]

CECILL License, http://www.cecill.info/licences/Licence_CeCILL_V1.1-US.html [Accessed: 7 April 2005, 11:43].

Center of Open Source and Government (eGovOS) http://www.egovos.org/ [Accessed: 7 April 2005, 13:31]

Connell, C., Open Source *Projects Manage Themselves? Dream On* IBM Lotus Developer Network Archives, http://www-10.lotus.com/ldd/devbase.nsf/articles/doc2000091200 [Accessed: 7 April 2005, 13:31]

Connor, D., 2001, *Linux Slips Slowly into the Enterprise Realm,* Network World Fusion, http://www.nwfusion.com/news/2001/0319specialfocus.html [Accessed: 7 April 2005, 13:31]

Conway, P., December 2003, *Zope at Duke University: Open Source Content Management in a Higher Education Context*, *DigiCULT.Info*, Issue 6, p. 10, http://www.digicult.info/pages/newsletter.php [Accessed: 7 April 2005, 11:57].

Creative Commons, http://creativecommons.org/ [Accessed: 7 April 2005, 11:57].

Dahdah, H., February 2003, Open Source *Library System a Welcome Gift,* Computer World, http://www.computerworld.com.au/index.php/id;534895878;relcomp;1 [Accessed: 7 April 2005, 13:31]

Danish Board of Technology, Open Source *Software in e-government* http://www.tekno.dk/pdf/projekter/p03_opensource_paper_english.pdf [Accessed: 7 April 2005, 13:31]

Dean, K., 2003, *Data Flood Feeds Need For Speed, Wired.com*, http://www.wired.com/news/infostructure/0,1377,57625,00.html [Accessed: 7 April 2005, 13:35]

The DigiCULT Report Full Report

Technological Landscapes for tomorrow's cultural economy: Unlocking the value of cultural heritage, pp.212. Available online at http://www.digicult.info/pages/report.php, [Accessed: 7 April 2005, 11:46]

Digitale Bibliothek, http://digbib.iuk.hdm-stuttgart.de/gsdl/cgi-bin/library [Accessed: 7 April 2005, 13:31]

Dravis, P., Open Source *Software: Perspectives for Development* http://www.infodev.org/symp2003/publications/OpenSourceSoftware.pdf [Accessed: 7 April 2005, 13:31]

Dyck,T., February 2002, *Server Databases Clash, eWeek,* http://www.eweek.com/article2/0,3959,293,00.asp [Accessed: 7 April 2005, 13:31]

enCore Open Source MOO project, http://lingua.utdallas.edu/encore/, [Accessed: 7 April 2005, 13:31]

EROS: An Open Source Multilingual Research System for Image Content Retrieval dedicated to Conservation-Restoration exchange between Cultural Institutions, http://www.c2rmf.fr/documents/c2r_eros.pdf [Accessed: 7 April 2005, 13:31]

European Schoolnet Virtual School: Software, Freeware and Shareware: www.eun.org/goto.cfm?sid=220 [Accessed: 7 April 2005, 13:31]

Forrester, J.E., B.P. Miller, 2000, *An Empirical Study of the Robustness of Windows NT Applications Using Random Testing*, ftp://ftp.cs.wisc.edu/paradyn/technical_papers/fuzz-nt.pdf [Accessed: 7 April 2005, 12:15].

Freshmeat.net http://freshmeat.net, Accessed: 7 April 2005, 13:31

Guercio, M. C. Cappiello, *File Formats Typology and Registries for digital preservation*, (DELOS, WP6 D6.3.1), http://www.dpc.delos.info [Accessed 7 April 2005, 11:48].

Ge, L., L. Scott, M.Vanderwiele, 17 December 2003, *Putting Linux Reliability to the Test,* IBM DeveloperWorks, http://www-106.ibm.com/developerworks/linux/library/l-rel/ [Accessed: 7 April 2005, 13:31]

Gilfillan, I., 16 December 2003, *PostgreSQL vs. MySQL:Which is Better?, Database Journal*, http://www.databasejournal.com/features/postgresql/article.php/3288951 [Accessed: 7 April 2005, 13:31]

Glance, D. G., J. Kerr,A. Reid, February 2004, *Factors Affecting the Use of Open Source Software in Tertiary Education Institutions, First Monday*,Volume 9, Number 2, http://www.firstmonday.org/issues/issue9_2/glance/ [Accessed: 7 April 2005, 13:31]

GNET, January 2000, *How Do Linux and Windows NT Measure Up in Real Life?, ID-side,* http://gnet.dhs.org/stories/bloor.php3 [Accessed: 7 April 2005, 13:31]

Haber, L., April 2002, *City Saves With Linux,Thin Clients, ZDNet.com,*

http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2860180,00.html [Accessed: 7 April 2005, 13:37]

The Halloween Documents http://opensource.org/halloween/ [Accessed: 7 April 2005, 13:31]

INFONOMICS Free/Libre and Open Source Software: Survey and Study http://www.infonomics.nl/FLOSS/report/ [Accessed: 7 April 2005, 13:31]

Insight Special Report: *Why Europe Needs Free and Open Source Software and Content in Schools*, 2004, http://www.eun.org/insight-pdf/special_reports/Why_Europe_needs_foss_Insight_2004.pdf [Accessed: 7 April 2005, 13:31]

Kaven, O., November 2001, *Performance Tests: File Server Throughput and Response Times, PC Magazine*, http://www.pcmag.com/article2/0,1759,16227,00.asp [Accessed: 7 April 2005, 13:31]

Lee, C., October 2001, Open Source*: A Promising Piece of the Digital Preservation Puzzle*. A slightly different version appears as *Open-Source Software: A Promising Piece of the Digital Preservation Puzzle*, Electronic Currents, Midwest Archives Conference (MAC) Newsletter, Volume 29, Number 2 (113), 26-28, http://www-personal.si.umich.edu/~calz/oss_preservation.htm [Accessed: 7 April 2005, 13:31]

Linux Migration.com, http://www.linuxmigration.com/ [Accessed: 7 April 2005, 13:31]

The Linux Weekly News http://lwn.net/ [Accessed: 7 April 2005, 13:31]

Manning, B. Bind, 2000, *Internet Usage Statistics*, http://www.isi.edu/~bmanning/in-addr-versions.html [Accessed: 7 April 2005, 13:31]

Mantarov, B., 1999 Open Source *Software as a New Business Model The Entry of Red Hat Software, Inc. on the Operating System Market with Linux*, Dissertation submitted in partial fulfilment of the degree of MSc in International Management at the University of Reading, http://bmantarov.free.fr/bojidar/academic/Dissertation_-_Open_source_software_as_a_new_business_model.pdf [Accessed: 7 April 2005, 13:31]

McMillan, R., 26 March 2004, *SCO Linux Licensee Has Second Thoughts on Deal, Computer World,* http://www.computerworld.com/governmenttopics/government/legalissues/story/0,10801,91671,00.html [Accessed: 7 April 2005, 13:31]

Munich Linux Decision Final, 14 June 2004, *DesktopLinux.com*, http://www.desktoplinux.com/news/NS7137390752.html [Accessed: 7 April 2005, 13:31]

NASA Open Source License, http://www.opensource.org/licenses/nasa1.3.php [Accessed 7 April 2005, 11:56].

Netcraft Web Server Survey, http://news.netcraft.com/ [Accessed: 7 April 2005, 13:31]

Norwegian Board of Technology Global CountryWatch on Open Source Policy, http://www.teknologiradet.no/html/592.htm [Accessed: 7 April 2005, 13:31]

OpenSector.org, Public Sector Related Content, http://opensector.org/ [Accessed: 7 April 2005, 13:31]

Open Source and Industry Alliance, http://www.osaia.org/ [Accessed: 7 April 2005, 13:31]

The Open Source Initiative, http://www.opensource.org [Accessed: 7 April 2005, 13:31]

Open Source Software Watch (OSS Watch), http://www.oss-watch.ac.uk/ [Accessed: 7 April 2005, 13:31]

Open Source Technology Group, http://www.ostg.com [Accessed: 7 April 2005, 13:31]

O'Reilly,T., May 2004, *The Open Source Paradigm Shift, Tim.Oreilly.com* , http://tim.oreilly.com/opensource/paradigmshift_0504.html [Accessed: 7 April 2005, 13:31]

Perens, B. *et al*., *Free Software Leaders Stand Together,* http://perens.com/Articles/StandTogether.html [Accessed: 7 April 2005, 13:31]

Perens, B., *Open Standards Principles and Practice,* http://perens.com/OpenStandards/Definition.html [Accessed: 7 April 2005, 13:31]

PHP Usage Statistics, http://www.php.net/usage.php [Accessed: 7 April 2005, 13:31]

PostgreSQL or MySQL http://www-css.fnal.gov/dsg/external/freeware/pgsql-vs-mysql.html [Accessed: 7 April 2005, 13:31]

Raymond, E. S., 1999-2004, *The Cathedral and the Bazaar*, http://www.catb.org/~esr/writings/cathedral-bazaar/ [Accessed: 7 April 2005, 13:31]

Raymond, E. S., version 4.4.7, 2003, *The Jargon File,* http://www.catb.org/~esr/jargon/ [Accessed: 7 April 2005, 13:31]

Raymond, E. S., 2000, *The Software Release Practice How-To,* http://www.tldp.org/HOWTO/Software-Release-Practice-HOWTO/index.html [Accessed: 7 April 2005, 13:31]

Ross, S, M. Donnelly, M. Dobreva, D. Abbott, A. McHugh, A. Rusbridge, 2005, *Digicult Technology Watch Report 3* http://www.digicult.info/pages/techwatch.php [Accessed: 7 April 2005, 11:30]

Ross, S, A. Gow, 1999, *Digital archaeology? Rescuing Neglected or Damaged Data Resources*, (London & Bristol: British Library and Joint Information Systems Committee), ISBN 1900508516, http://www.ukoln.ac.uk/services/elib/papers/supporting/pdf/p2.pdf [Accessed: 7 April 2005, 11:47]

Rusbridge, A., April 2004, *XENA: Electronic Normalising Tool*, DigiCULT.Info, Issue 7,

page 32, http://www.digicult.info/pages/newsletter.php [Accessed 7 April 2005, 11:49]

Science Commons, http://science.creativecommons.org [Accessed: 7 April 2005, 11:57].

Shankland, S., M. Kane, R. Lemos, 30 October 2001, *How Linux Saved Amazon Millions, CNET news.com,* http://news.com.com/2100-1001-275155.html?legacy=cnet&tag=owv Sourceforge.net http://sourceforge.net [Accessed: 7 April 2005, 13:31]

Stallman, R. M. *et al*. The Free Software Foundation, http://www.fsf.org [Accessed: 7 April 2005, 13:31]

Stallman, R. M. Stallman.org, http://www.stallman.org [Accessed: 7 April 2005, 13:31]

Staples, T., April 2003, *The Fedora Project: An Open-Source Digital Object Repository Management System*, *D-Lib Magazine*, Volume 9, Number 4, http://www.dlib.org/dlib/april03/staples/04staples.html [Accessed: 7 April 2005, 12:15].

Stone, B., 2004, *The Linux Killer*, http://www.wired.com/wired/archive/12.07/linux.html?pg=4&topic=linux&topic_set=(none) [Accessed: 7 April 2005, 13:31]

Thurgood, A., January 2005, *The GPL and non-U.S. law*, Open Source *Law Blog*, http://www.oslawblog.com/2005/01/gpl-and-non-us-law.html [Accessed: 7 April 2005, 11:43].

Trimble, P.S., December 2000, *Open Minds on Open Source*, *FCW.com*, http://www.fcw.com/fcw/articles/2000/1204/pol-nasa-12-04-00.asp [Accessed: 7 April 2005, 11:57].

Wallen Jr, J., 29 November 2002, *OpenOffice.org versus Microsoft Office, ZDNet.com Australia* on http://www.zdnet.com.au/insight/0,39023731,20270300,00.htm [Accessed: 7 April 2005, 13:31]

Wheatley, P., 2004, *Institutional Repositories in the context of Digital Preservation DPC Technology Watch Series* Report 04-02, http://www.dpconline.org/docs/DPCTWf4word.pdf [Accessed: 22 July 2005, 14:35]

Wheeler, D.A., Rev. 2005, *Why Open Source Software / Free Software (OSS/FS, FLOSS or FOSS)? Look at the Numbers!*, http://www.dwheeler.com/oss_fs_why.html [Accessed: 7 April 2005, 13:31]

Windows 2000 Versus Linux in Enterprise Computing An Assessment of Business Value for Selected Workloads http://www.microsoft.com/windows2000/docs/TCO.pdf [Accessed: 7 April 2005, 13:31]

Witten, I. H., D. Bainbridge, S. J. Boddie, October 2001, *Greenstone Open Source Digital Library Software, D-Lib Magazine*, Volume 7, Number 10, http://www.dlib.org/dlib/october01/witten/10witten.html [Accessed: 7 April 2005, 13:31]

Fora

Association of C and C++ Users (ACCU) Open Source Forum 2004, 14-17 April 2004, Oxford, England
http://www.reportlab.com/conferences/accu2004/index.html

Glossary of Terms

Creative Commons - A non-profit organisation devoted to expanding the range of creative work available for others to legally build upon and share.

Curatability - A measure of the ease with which a digital resource can be curated.

Distribution - A software release representing a packaging up of several individual programs; most commonly a version of GNU/Linux with associated applications.

Emulation - The process of recreating existing hardware or software environments with software.

Free Software - Software released under terms conforming to the four fundamental freedoms established by the Free Software Foundation, which demand transparency and the legal and practical freedoms to change, re-use and re-distribute code.

Freedom of Information - UK legislation compelling public bodies to release information on request.

Institutional Repository - A software infrastructure designed to store digital resources for the facilitation of their management and/or preservation.

Migration - The process of moving digital resources to alternative hardware or software environments to facilitate their use in the face of obsolescence and ensure their longevity.

Proprietary Software - Examples of software where the user cannot control functionality or study or edit the code.

Representation Information - The information that maps a Data Object into more meaningful concepts. An example is the ASCII definition that describes how a sequence of bits (i.e., a Data Object) is mapped into a symbol. In order to keep things manageable, Representation Information can be factored in distinct types, such as structure, semantics and others. The latter can include software and standards, among other things. This normalisation allows one, for example, to describe two sets of information which are identical, but which are held in different structures (formats), by combining the same Semantic description with different Structure descriptions.

Software License - An agreement distributed alongside computer software determining acceptable legal use for that software.

Source Code - Pre-compiled, human-readable program code.

Standard - An accepted practice, technology or specification.

Total Cost of Ownership - The financial costs associated with a particular activity or

policy, incorporating all costs incurred, including acquisition, maintenance, staff and retraining costs.

Acronyms and Abbreviations

BBC - British Broadcasting Corporation.

BIOS - Biological Innovation for Open Source.

BSD - Berkeley System Design.

CVS - Concurrent Version System.

ebXML - Electronic Business Using XML.

FAQ - Frequently Asked Questions.

FLOSS - Free, Libre or Open Source Software.

FS - Free Software.

FSF - Free Software Foundation.

FUD - Fear, Uncertainty and Doubt.

GNU - GNU's Not Unix.

GPL - General Public License.

HTML - Hypertext Markup Language.

IBM - International Business Machines.

LOCKSS - Lots of Copies Keeps Stuff Safe.

METS - Metadata Encoding and Transmission Standard.

OSI - Open Source Initiative.

OSS - Open Source Software.

PDF - Portable Document Format.

PHP - PHP Hypertext PreProcessor.

Perl - Practical Extraction and Report Language.

SQL - Structured Query Language.

TCO - Total Cost of Ownership.

VLE - Virtual Learning Environment.

WINE - Wine is Not an Emulator.

XENA - XML Electronic Normalising of Archives.

XML - eXtensible Markup Language.

About the Author

Since graduating with a Scots Law Degree with Honours from Glasgow University in 2000, Andrew McHugh has concentrated on developing a wide range of skills mainly in the digital realm. He collected his Masters in Information Technology, again from Glasgow University in 2001 and since then has been employed within HATII (the Humanities Advanced Technology and Information Institute) at this University in various capacities. Within the institution's Department of Music he revolutionised the information infrastructure, applying and honing several skills and performing a diverse range of roles, with responsibilities ranging from database and server administration to web programming, application development and desktop cluster design and management. In late 2004 he joined the Digital Curation Centre in the position of Advisory Services Manager, leading a world-class team of digital curation practitioners in offering leading-edge expertise and insight in a range of issues to a primarily HE and FE audience. In his spare time Andrew maintains an aptitude and enthusiasm for software development and continues to develop web-based solutions for a range of customers including commercial and heritage clients. He is a keen user of open source technologies with several GNU/Linux distributions including Fedora Core, Gentoo and SUSE distributed across the hard disk partitions of his various systems, including a Microsoft X-Box games console.